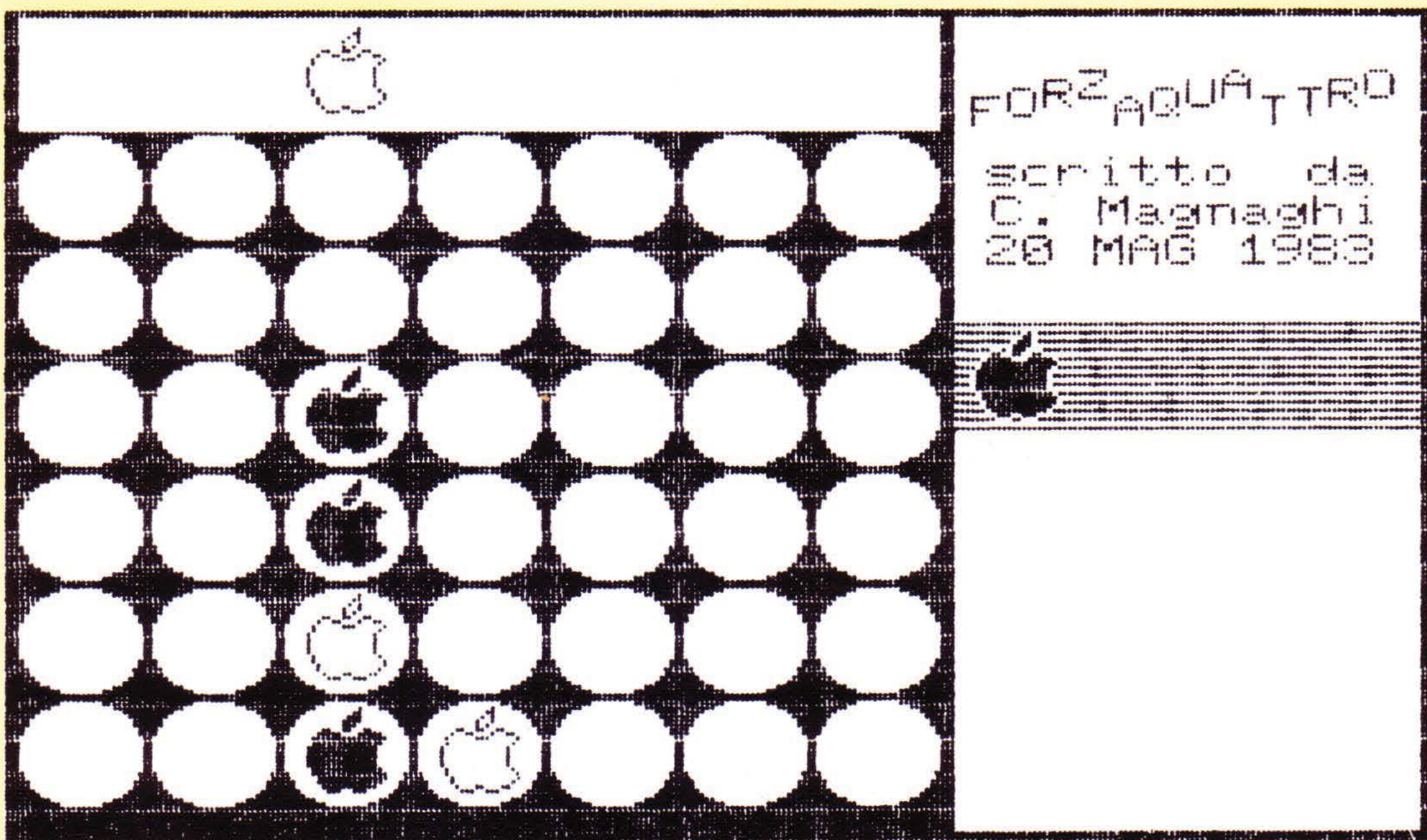
Bit RISERVATO PERSONAL

Spedizione in Abb. Postale Gruppo III/70
Questo fascicolo è annesso alla rivista Bit n. 46 Gennaio 1984

Forza quattro! Dalla TV un gioco "esperto" per Apple

L'idea di scrivere questo programma è nata dal desiderio di sperimentare le tecniche solitamente usate nei programmi che giocano a scacchi in un caso sufficientemente semplice da poter utilizzare un linguaggio ad alto livello.

Figura 1 - La videata di una partita di Forzaquattro, *vincente se bene opereremo ...*

di **C. Magnaghi**

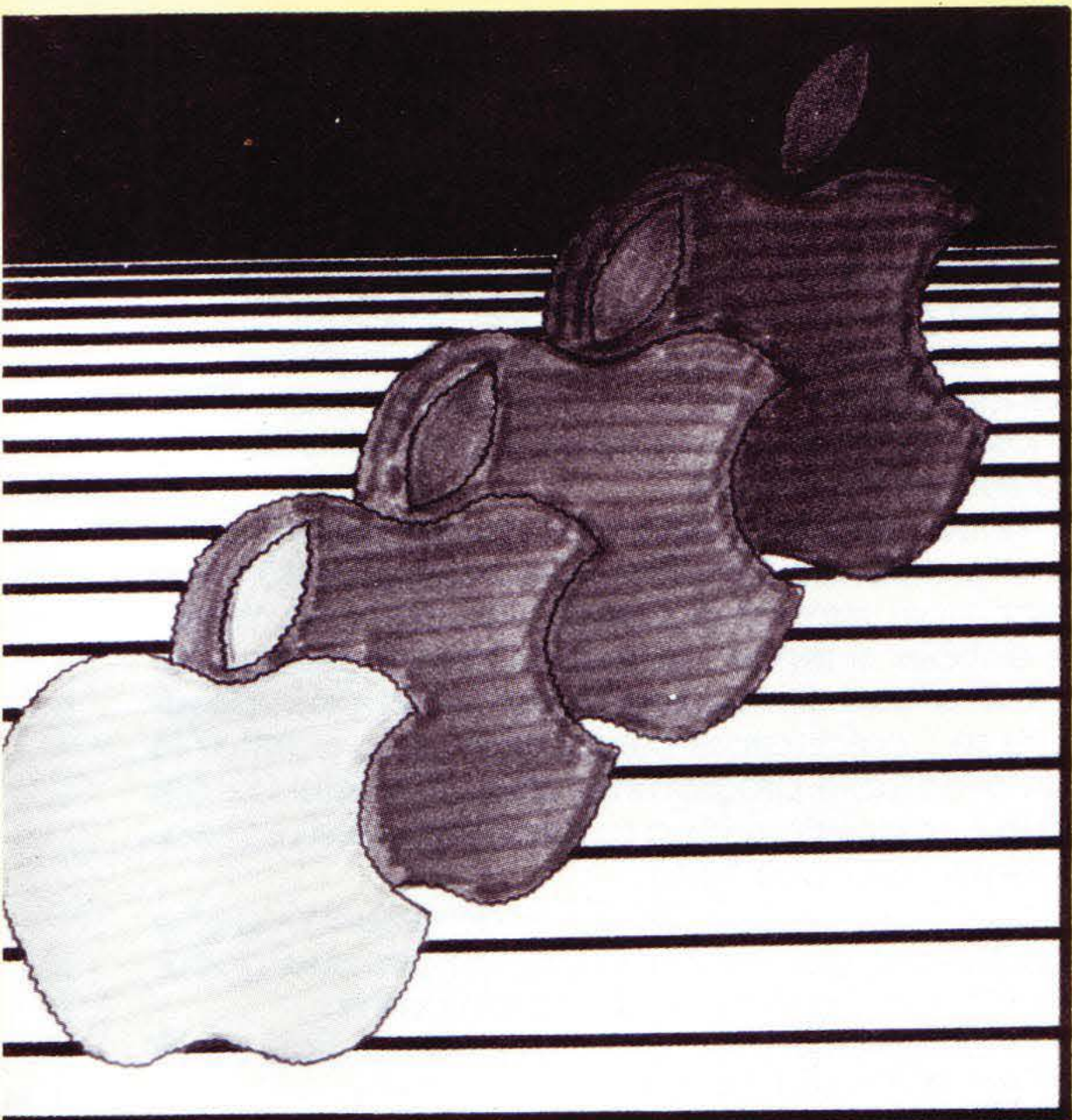
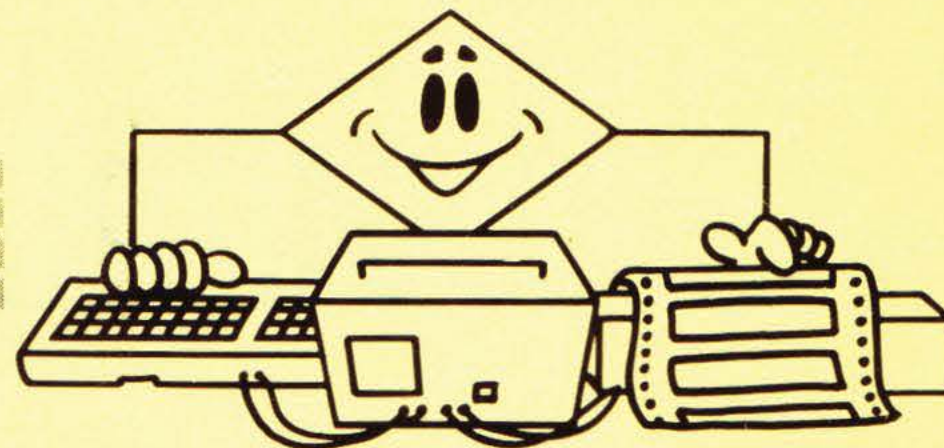
I compilatori producono infatti un codice macchina (o un P-code che deve essere ulteriormente interpretato nel caso del Pascal) particolarmente inefficiente rispetto a un assembler, ma offrono il notevole vantaggio, per chi vuole provare delle particolari tecniche di programmazione, di un codice sorgente sufficientemente facile da leggere.

Il gioco forzaquattro si è dimostrato particolarmente adatto a questo scopo; infatti, pur essendo sufficientemente stimolante, presenta un albero di gioco con un indice di diramazione (numero di mosse possibili in una data posizione) particolarmente basso (sono possibili al più sette mosse mentre, per esempio negli scacchi il numero tipico di mosse possibili si aggira sulla ventina) e tuttavia la notevole profondità dell'albero di gioco impedisce una analisi esaustiva delle situazioni possibili.

L'importanza dell'indice di diramazione è particolarmente importante per quanto riguarda il tempo di calcolo, infatti è evidente che il numero di rami terminali presenti in un albero con indice di diramazione D e con una profondità H è di D∧H e, pur potendo ridurre il numero di rami terminali da considerare a 2★D∧(H/2), è praticamente impossibile scrivere in un linguaggio ad alto livello e su un piccolo elaboratore un programma che giochi, per esempio a scacchi, analizzando l'albero di gioco fino a una profondità, per esempio, di quattro livelli.

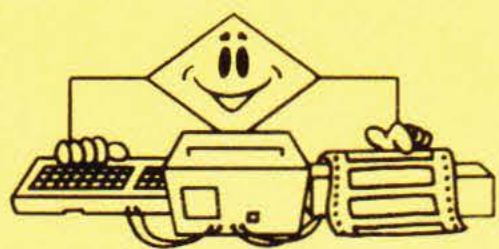
Il metodo usato per l'analisi dell'albero è quello dei minimi e massimi (si tratta quindi di un tipo di ricerca su alberi di tipo ben diverso da quello descritto da Gregg Williams nell'articolo tradotto in italiano su Personal Software) e per ridurre il numero di rami terminali da analizzare è stato usato l'algoritmo alfa-beta senza ordinamento (da prove effettuate è risultato che il tempo impiegato per ottimizzare l'ordine di valutazione dei rami non è compensato da un guadagno nell'efficienza dell'algoritmo dei minimi-massimi in un albero tanto ridotto). Ne è risultato un programma sufficientemente abile nel gioco, ma è emerso nel corso di varie prove che il gioco Forzaquattro presenta singolari strutture per le quali in una delle prime mosse è possibile forzare partite che porteranno alla vittoria verso la quarantesima mossa.

Per esempio consideriamo la partita in figura 1, credo possa essere utile al fine della comprensione del problema dimostrare che esiste una mos-

sa che assicura al bianco la vittoria molto più tardi nella partita, darò quindi più tardi la soluzione di questo problema.
Credo comunque sia assurdo sperare che una analisi svolta a una profondità limitata sia in grado di rivelare simili situazioni. Non resta quindi che munire il programma di una libreria di mosse sconsigliate.
Prima di descrivere con maggior dettaglio le tecniche usate nel programma vorrei motivare la scelta del linguaggio e spiegare come gli utenti con singolo drive che volessero cimentarsi nell'impresa possano scrivere compilare e linkare il programma (per gli utenti con più di un drive non dovrebbero sussistere problemi).

Il primo linguaggio che può venire in mente di usare sull'Apple è certamente il BASIC, ma questo linguaggio presenta vari difetti: per prima cosa è lento, a meno che non lo si compili, poi mancando della possibilità di definire sottomoduli ha una forte tendenza a trasformare programmi (la cui struttura è già poco chiara nella mente dell'autore che sta cercando di sperimentare delle idee e che quindi andrà incontro a molte successive revisioni) in masse amorfe di istruzioni da cui risulta poi impossibile discernere la struttura del programma stesso. Inoltre, pur essendo possibile scrivere processi ricorsivi, le strutture di programma e di dati che bisogna inserire per fare ciò che in altri linguaggi (LISP e Pascal) è del tutto naturale, finirebbero per soffocare completamente la struttura del programma il che spingerebbe prima o poi l'autore a rinunciare all'impresa. I motivi che mi hanno indotto a rinunciare al BASIC portano anche a scartare il FORTRAN nel quale inoltre è stato fatto ogni sforzo per rendere illegale l'uso della ricorsione, il linguaggio macchina era già stato escluso perchè incompatibile con i fini del programma (ma rientrerà più tardi risolvendo trionfalmente una situazione che sembrava senza speranza), il FORTH ha il difetto di avere un codice sorgente praticamente illeggibile ed è quindi poco adatto a un programma soggetto a successive riletture. Il LISP pur essendo un linguaggio a volte simpatico da usare non ha istruzioni di sistema per utilizzare le matrici (l'idea era di memorizzare la scacchiera come una matrice), e cercare di gestire le matrici come liste di liste da cui estrarre gli elementi con le istruzioni CAR e CDR è un processo troppo lungo in termini di tempo per un programma che mostra una forte tendenza a divenire un manipolatore di matrici.

Non resta quindi che il Pascal (non ho a disposizione il Modula 2, un linguaggio che avrebbe, almeno sulla carta, brillantemente risolto alcuni problemi incontrati nella scrittura del programma). Coloro che, utenti di singolo drive, ben conoscono i problemi dell'editazione di un programma di grosse dimensioni si domanderanno quanto spazio occupa su dischetto il codice sorgente del programma che stò descrivendo, non impressionatevi, solo ventotto settori. Abbastanza per non stare sull'Apple0 insieme al suo codice oggetto. Il motivo per cui non dovete preoccuparvi è che parecchio spazio può essere liberato su detto dischetto in caso di necessità: infatti il file SYSTEM.SYNTAX è del tutto superfluo (basta tener aperta la pagina 137 del manuale) e il SYSTEM.FILER può essere lanciato indifferentemente da qualsiasi dischetto (basta poi far attenzione a rimettere nel drive il disco di bootstrap prima di abbandonare il filer). Cancellando questi due file dall' Apple0 c'è spazio sufficiente sul dischetto per editare in tutta tranquillità: basta poi assemblare la funzione VALUE, trasferire il codice oggetto su Apple0, lanciare il LINKER e seguire le istruzioni.

È inoltre necessario prima di poter utilizzare il programma, compilare e far girare un programma ausiliario denominato MAKEPIC che crea un file denominato PICTURES contenente le immagini che verranno utilizzate dal programma principale.Consiglio inoltre di non inserire in un primo momento l'opzione R- nel programma, in quanto questo potrebbe contenere degli errori che senza questa opzione sono più facili da trovare, potrete poi ricompilare il programma con questa opzione per farlo girare leggermente più veloce.

Il cuore del programma è la funzione RVALUE (dove R sta per ricorsivo), questa funzione sviluppa l'analisi dell'albero di gioco fino a una profondità che viene determinata dalla funzione chiamante CHOICE a seconda dell'indice di diramazione previsto per l'albero di gioco (questo per approfittare delle situazioni più semplici analizzandole più a fondo) con il già citato metodo dei minimi-massimi. Arrivata alla massima profondità RVALUE invece di richiamare se stessa chiama la funzione VALUE che essendo la parte di programma maggiormente usata è stata scritta in linguaggio macchina per ridurre i tempi di calcolo (i risultati di questa scelta sono stati entusiasmanti: scrivendo in codice macchina questa piccola porzione di programma il tempo di elaborazione è sceso da circa cinque minuti a circa venti secondi per le mosse più complesse). RVALUE calcola poi il massimo o il minimo (a seconda della parità del livello a cui chiama la funzione VALUE) dei valori per le varie colonne e assegna questo valore alla posizione. Il valore della mossa viene quindi calcolato ricorsivamente.
Cercherò di essere più chiaro con un esempio: supponiamo che il calcolatore stia cercando

APPLE

che valore ha per lui una certa mossa A: qual'è il valore di A? E' il minimo dei valori delle contromosse possibili (infatti l'avversario cercherà di minimizzare il suo guadagno : questo è il concetto che stà alla base del metodo dei minimi-massimi); analogamente il valore di ciascuna contromossa è il massimo dei valori delle contro - contromosse possibili. È chiaro come questa definizione generi un processo ricorsivo.

Ma abbiamo detto all'inizio che questo procedimento da solo è inadeguato.

Per spiegare come il calcolatore risolva questo problema introdurrò da prima una notazione, che tra l'altro è la stessa utilizzata dal calcolatore per analizzare le partite.

E' ovvio da come è definito il gioco che per indicare una mossa è sufficiente indicare la colonna in cui si intende muovere, una notazione compatta è quindi la seguente: C 33323...

intendendo con C indicare che ha mosso per primo il calcolatore, col primo 3 che ha mosso nella terza colonna, col secondo che anche il giocatore ha mosso nella terza colonna e così via.

Supponiamo ora che il calcolatore dopo aver memorizzato la partita fino a un certo punto, diciamo C 323562 (i numeri sono messi a caso) si accorga di non avere più alcuna possibilità; allora dovrà concludere che la mossa da lui fatta precedentemente era sbagliata; metterà quindi nel suo libro nero la seguente partita C 32356; ora guarda il libro nero e trova che di fianco a questa ve ne sono altre simili e che la pagina del libro nero risulta:

C 32356
C 32355
C 32354

Ora nota che le colonne 1 e 2 erano impraticabili (perchè piene) e che in un altro taccuino aveva segnato che 2 colonne venivano già segnalate dall'algoritmo dei minimi-massimi come perdenti, ma 3+2+2=7 e quindi non ci sono proprio mosse accettabili.

Questo significa che la mossa precedente era errata e quindi il calcolatore sostituirà alle partite presenti nella pagina la partita C 323.

Si noti che questa nuova informazione non è solo un riassunto delle precedenti per risparmiare spazio, ma è una sintesi costruttiva; infatti ora il calcolatore che mentre gioca guarda il libro nero si renderà conto con una mossa di anticipo che stà andando in un vicolo cieco.

Il processo descritto viene iterato fin quando è possibile, cercando di sintetizzare al massimo le informazioni presenti in memoria ogni volta che ne viene inserita una nuova: tale lavoro viene svolto dal sottoprogramma LEARN e utilizzato dalla funzione BADGAME.

Naturalmente il calcolatore può impiegare un numero di partite inaccettabilmente elevato per capire qual'è l'errore effettivo, se un errore che provoca la sconfitta quando il tabellone è completamente pieno si verifica nelle primissime mosse; a questo scopo è stato inserito il comando I per segnalare al calcolatore che la mossa appena eseguita è debole; con questo comando, dopo aver chiesto conferma, il calcolatore inserisce la partita fin qui memorizzata nel libro nero; è naturalmente vostra responsabilità far si che le mosse segnalate come deboli lo siano realmente; inoltre questo tipo di inserimento non cancellerà necessariamente le informazioni divenute superflue: per esempio l'inserimento di C 13574 può non cancellare un già presente C 13574574 (i numeri sono sempre messi a caso).

Vorrei qui notare che, utilizzando il calcolatore il generatore di numeri casuali solo per la prima mossa non sarebbe necessario ricordare anche le mosse che lui ha fatto (eccetto la prima) essendo queste implicitamente determinate dalle mosse fatte dal giocatore ed eventualmente dalla sua prima mossa.

Il motivo per cui questo metodo non è stato usato è che, essendo stata ammessa la possibilità che il giocatore segnali al calcolatore che una mossa è debole, si rende necessario creare dei dati espliciti e facili da rivedere; questo sarebbe impossibile se il calcolatore per mostrare una partita (in un eventuale programma atto all'analisi del file che il calcolatore crea su disco) dovesse praticamente rigiocarla tutta.

Vorrei concludere descrivendo alcuni bug che ho trovato nel Apple Pascal o nella sua documentazione sviluppando il programma:

1) il parametro yskip della procedura drawblock (pag. 96) non funziona.

2) Drawblock e fillscreen sembrano utilizzare in modo diverso i parametri definiti da viewport: in particolare drawblock non scrive nell'ultima colonna liberata da un comando del tipo fillscreen (black).

3) Il manuale del sistema operativo mostra a pagina 255 che il Pascal data heap parte dalla locazione $C00; in realtà usando la grafica questo non è più vero e il data heap parte da $4000

4) Gli operatori >, <, <=, >= non sono utilizzabili con gli array (questo a onor del vero è riportato sul manuale in modo implicito quando a pagina 85 questi operatori non vengono riportati insieme a < > e a =.

Questi problemi del tutto secondari non possono tuttavia modificare il mio giudizio più che positivo sull'Apple Pascal come linguaggio di uso generale.

Per concludere un plauso a una innovazione che il Modula 2 porta rispetto al Pascal: l'elaborazione condizionale delle espressioni logiche; questo significa che A OR B se A è vero può essere calcolato e vale vero qualunque sia il valore di B anche se questo non è definito, un caso tutt'altro che raro quando si opera con variabili dinamiche.

Ah, quasi dimenticavo; la soluzione al problema proposto è muovere nella colonna 4 e successivamente muovere ogni volta nella stessa colonna in cui ha mosso il nero: è facile vedere che in questo modo prima o poi si vince comunque.

Il gioco

Il gioco Forzaquattro è stato pubblicizzato pochi mesi or sono in televisione ed è diventato abbastanza conosciuto e diffuso.

Si gioca su un tabellone verticale formato da sei righe e sette colonne; le pedine vengono lasciate cadere dall'alto e vanno a posarsi nella casella libera più in basso della colonna scelta.

Scopo del gioco è riuscire ad allineare quattro pedine in orizzontale, verticale o diagonale pri-

ma dell'avversario. È possibile che dopo aver riempito tutto il tabellone nessuno dei due giocatori sia riuscito nell'intento: in tal caso la partita risulta patta.
Nella versione per calcolatore il tabellone di gioco viene mostrato sul monitor e il calcolatore chiede al giocatore chi deve giocare per primo.
Sono possibili tre risposte:
Utilizzatore;
Calcolatore;
Abbandono;
questa ultima opzione è disponibile sempre in fase di input ad eccezione di quando il calcolatore chiede conferma di un comando dato dall'utente e sostituisce il classico CTRL-@ che non deve più essere utilizzato, in quanto dopo ogni seduta di gioco il calcolatore deve salvare su dischetto le nuove informazioni elaborate sul gioco stesso.

Quando il giocatore deve muovere, compare sullo schermo in alto un segno indicante la colonna in cui ha mosso il calcolatore; questo segno agisce poi come cursore per i comandi successivi:
< (senza premere lo SHIFT) sposta il cursore a sinistra.
> sposta il cursore a destra.
<SPACE> lascia cadere la pedina nella colonna su cui è posizionato il cursore.
"Abbandona" esce definitivamente dal programma.
"Insegna" segnala al calcolatore che l'ultima mossa da lui fatta è debole, il calcolatore chiede conferma e poi riprende la fase di input continuando la partita.
"Riparti" termina la partita e ritorna nella condizione iniziale.
"Suono" attiva o disattiva le subroutine di suono del calcolatore (particolarmente utile quando si gioca in ufficio ...)

I messaggi del calcolatore

Nella presente versione i messaggi del calcolatore sono in italiano. Personalmente preferisco i messaggi in lingua inglese; nel programma ho quindi lasciato scritti i messaggi inglesi equivalenti come commento.

Chi preferisce questa seconda versione non deve far altro che sostituire a ciò che compare nel programma quello che compare di fianco o comunque nelle immediate vicinanze come commento; l'unica eccezione è il terz'ultimo commento della procedura start:

```
(★MOVE(1);
WCHAR (CHR(1));★)
```

che va solamente inserito.

Bibliografia

- Byte October 1978
- Gödel, Escher, Bach an Eternal Golden Braid
Douglas R. Hofstadter - Vintage Books New York 1980

(Ringraziamo Informatica Shop di Milano per il temporaneo sfruttamento della stampante Silentype). ■

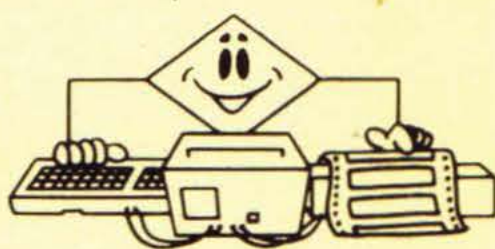
Figura 2 - Il listato.

```
(*$L CONSOLE:*)
PROGRAM makepic;

VAR
 f:FILE OF PACKED ARRAY[1..24,1..48]OF boolean;

PROCEDURE conv(row:integer;str48:string);

VAR i:integer;

BEGIN
 FOR i:=1 TO 48 DO
  f^[row,i]:=(str48[i]='*')
END;


PROCEDURE sub1;

BEGIN
 conv(24,'              *         ********        ********');
 conv(23,'             *:*        ******            ******');
 conv(22,'            *::*        ****                ****');
 conv(21,'           *:::*        ***                  ***');
 conv(20,'          *::::*        **                    **');
 conv(19,'       ** *:::***       **                    **');
 conv(18,'      *::**::**::**     *                      *');
 conv(17,'     *:::::***:::::*    *                      *');
 conv(16,'    *:::::::::::::::*                           ');
 conv(15,'    *::::::::::::::*                            ');
 conv(14,'   *::::::::::::::*                             ');
 conv(13,'   *::::::::::::::*                             ')
END;
```

Seguito figura 2.

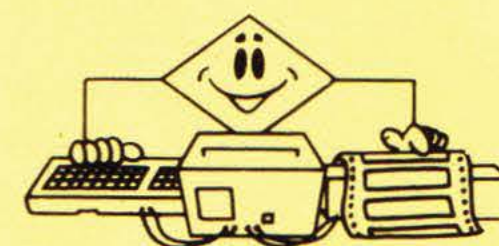
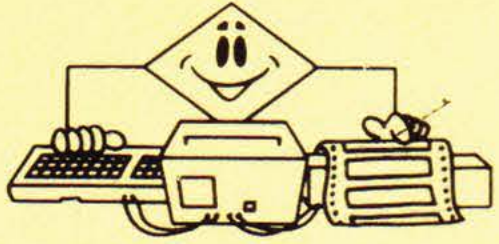
```
PROCEDURE sub2;

BEGIN
 conv(12,'   *::::::::::::::*                             ');
 conv(11,'   *::::::::::::::*                             ');
 conv(10,'   *::::::::::::::*                             ');
 conv( 9,'    *::::::::::::::*                            ');
 conv( 8,'    *:::::::::::::::*   *                      *');
 conv( 7,'     *::::::::::::::*   *                      *');
 conv( 6,'     *:::::::::::::*    **                    **');
 conv( 5,'      *:::::::::::*     **                    **');
 conv( 4,'       *:::**:::**      ***                  ***');
 conv( 3,'        ***  ***        ****                ****');
 conv( 2,'                        ******            ******');
 conv( 1,'                        ********        ********')
END;

PROCEDURE sub3;

BEGIN
 conv(24,'                                                ');
 conv(23,'               *                      *         ');
 conv(22,'              **                     **         ');
 conv(21,'             ***                    * *         ');
 conv(20,'            ****                   *  *         ');
 conv(19,'            ***                    * *          ');
 conv(18,'        **  **  **             **  **  **       ');
 conv(17,'       *****   *****          *  **   *  **     ');
 conv(16,'      ***************        *     ***     *    ');
 conv(15,'      **************         *            *     ');
 conv(14,'     **************         *            *      ');
 conv(13,'     **************         *            *      ')
END;

PROCEDURE sub4;

BEGIN
 conv(12,'     **************         *            *      ');
 conv(11,'     **************         *            *      ');
 conv(10,'     **************         *            *      ');
 conv( 9,'      **************         *            *     ');
 conv( 8,'      ***************        *             *    ');
 conv( 7,'       **************         *            *    ');
 conv( 6,'       *************          *           *     ');
 conv( 5,'        ***********            *   **   **      ');
 conv( 4,'         ***  ***               ***  ***        ');
 conv( 3,'                                                ');
 conv( 2,'                                                ');
 conv( 1,'                                                ')
END;
```

```
BEGIN
 sub1;sub2;
 rewrite(f,'PICTURES');
 put(f);
 sub3;sub4;
 put(f);
 close(f,LOCK)
END.
(*$C copyright Carlo Magnaghi *)
(*$S+,G+,R-*)
(*$L CONSOLE:*)
PROGRAM fourinarow;
USES TURTLEGR,APPLESTU;

CONST
 compudot=5;
 humandot=-5;
 emptydot=0;
 outdot=maxint;
 fourline=500;
 illegal=-10000;

TYPE
 gametrack=PACKED ARRAY[1..40]OF 0..7;
```

APPLE

Seguito figura 2.

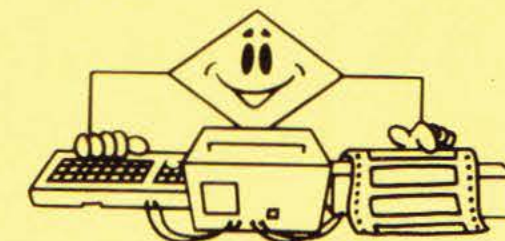
```
mover1=(FH,FC);
matrix=ARRAY[0..7,0..8] OF integer;
link=^game;
game=PACKED RECORD
        move:gametrack;
        first:mover1;
        next:link
       END;

VAR
 apple:PACKED ARRAY[1..24,1..48]OF boolean;
 aux    :PACKED ARRAY[1..24,1..48]OF boolean;
 board : matrix;
 i,j : integer;
 keyin : CHAR;
 time : integer;
 sound : boolean;
 base1,base,basec,baseh:link;
 movenum:integer;
 trace:gametrack;
 starter:mover1;
 mirror:integer;
 notset:boolean;
 learned:boolean;
 status:(full,learning,tracing);
 numcol:ARRAY[0..21]OF integer;
 quit:boolean;
 maxdepth:integer;


PROCEDURE play;FORWARD;


PROCEDURE home;

VAR i:integer;

BEGIN
 pencolor(none);chartype(0);
 FOR i:=0 TO 2 DO
  BEGIN
   moveto(192,22+8*i);
   wstring('                    ')
  END
END;


FUNCTION gt (alpha,beta:gametrack):boolean;

VAR i:integer;

BEGIN
 FOR i:=1 TO 40 DO
  IF alpha[i]>beta[i] THEN
   BEGIN gt:=true;EXIT(gt) END;
  gt:=false
END;


PROCEDURE sound0;

BEGIN
 IF sound THEN
  BEGIN
   note(7,20);
   note(22,20);
   note(31,20);
   note(36,25);
   note(0,50);
   note(31,30);
   note(36,125)
  END
END;
```

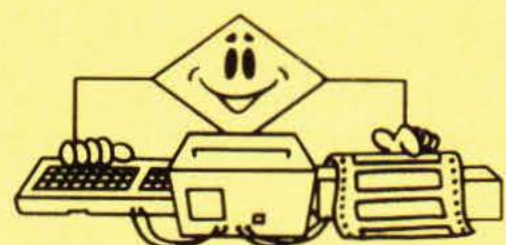
Seguito figura 2.

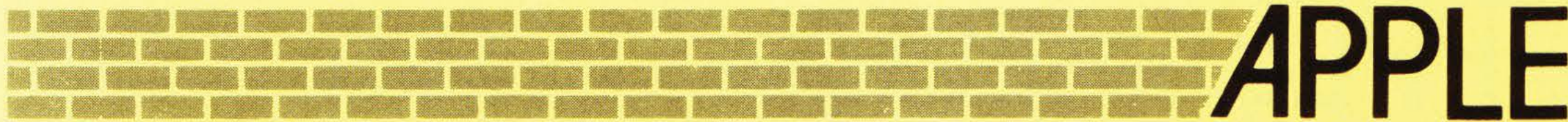
```
PROCEDURE sound1;

VAR i,j:integer;

BEGIN
 IF sound THEN
  BEGIN
   FOR i:=1 TO 30 DO
    BEGIN
     FOR j:=1 TO random MOD 20 DO;
     note(1,1)
    END;
   FOR i:=1 TO 20 DO
    BEGIN
     FOR j:=1 TO 20*i DO;
     note(1,1)
    END
  END
END;



PROCEDURE sound2;
BEGIN
 IF sound THEN
  BEGIN
   note(26,35);
   note(26,35);
   note(28,35);
   note(30,35);
   note(26,35);
   note(30,35);
   note(28,35);
   note(21,35);
   note(26,35);
   note(26,35);
   note(28,35);
   note(30,35);
   note(26,70);
   note(25,35);
   note(21,35);
   note(26,35);
   note(26,35);
   note(28,35);
   note(30,35);
   note(31,35);
   note(30,35);
   note(28,35);
   note(26,35);
   note(25,35);
   note(21,35);
   note(23,35);
   note(25,35);
   note(26,70);
   note(26,35)
  END
END;



PROCEDURE start1;

VAR row,column:integer;

BEGIN
 viewport(3,182,164,189);fillscreen(black);
 viewport(0,279,0,191);
 FOR row:=0 TO 5 DO
 FOR column:=0 TO 6 DO
  drawblock(aux,6,24,0,24,24,3+26*column,9+26*row,10)
END;



PROCEDURE start;
```

```
VAR
 top:string;
 i:integer;
 pictures:FILE OF PACKED ARRAY[1..24,1..48]OF boolean;

BEGIN
 reset(pictures,'PICTURES');
 aux:=pictures^;
 get(pictures);
 apple:=pictures^;
 close(pictures);
 initturtle;
 fillscreen(white);
 viewport(186,276,3,189);
 fillscreen(black);
 viewport(186,277,3,189);
 pencolor(none);moveto(189,164);chartype(10);
 top:='FORZAQUATTRO';(*top:='FOURIN AROW';*)
 FOR i:=1 TO 12 DO(*FOR i:=1 TO 11 DO*)
  BEGIN
   wchar(top[i]);
   turnto(90);move(2);
   IF (i=4)OR(i=8) THEN
    BEGIN
     turnto(-90);move(8)
    END
  END;
 (*move(1);
 wchar(chr(1));*)
 moveto(192,150);
 wstring('scritto  da');(*wstring('written  by');*)
 moveto(193,141);
 wstring('C. Magnaghi');
 moveto(192,132);
 wstring('20 MAG 1983');(*wstring('20 MAY 1983');*)
 FOR i:=120 DOWNTO 95 DO
  IF odd(i) THEN
  drawblock(i,15,0,0,100,1,186,i,15);
 drawblock(aux,6,0,0,24,24,186,95,4);
 drawblock(apple,6,0,0,24,24,186,95,14);
 start1
END;



PROCEDURE remember;

VAR
 fst:boolean;
 aux,aux1:link;
 errcode:integer;
 memory:FILE OF game;

BEGIN
 baseh:=nil;basec:=nil;
 (*$I-*)
 reset(memory,'LOST.GAMES');
 errcode:=ioresult;
 IF (errcode=10) THEN BEGIN close(memory);EXIT(remember) END;
 IF errcode<>0 THEN BEGIN writeln('I/O error ',errcode);EXIT(PROGRAM) END;
 (*$I+*)
 new(aux);fst:=true;
 WHILE (NOT(eof(memory))AND(memory^.first=FC)) DO
  BEGIN
   IF fst THEN BEGIN fst:=false;basec:=aux END;
   aux^:=memory^;
   aux1:=aux;new(aux):aux1^.next:=aux;
   get(memory)
  END;
 IF NOT(fst) THEN BEGIN fst:=true;aux1^.next:=nil END;
 WHILE NOT(eof(memory)) DO
  BEGIN
   IF fst THEN BEGIN fst:=false;baseh:=aux END;
   aux^:=memory^;
   aux1:=aux;new(aux);aux1^.next:=aux;
```

Seguito figura 2.

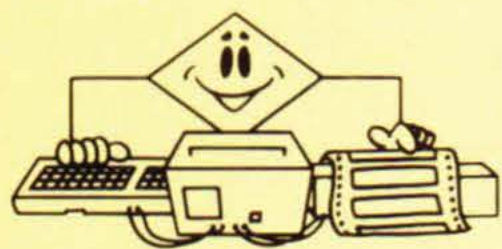
APPLE

Seguito figura 2.

```
    get(memory) ;
   END;
 IF NOT(fst) THEN aux1^.next:=nil;
 close(memory)
END;
PROCEDURE memorize;

VAR
 aux:link;
 memory:FILE OF game;

BEGIN
 IF learned THEN
  BEGIN
   rewrite(memory,'LOST.GAMES');
   aux:=basec;
   WHILE aux<>nil DO
    BEGIN
     memory^:=aux^;
     put(memory);
     aux:=aux^.next
    END;
   aux:=baseh;
   WHILE aux<>nil DO
    BEGIN
     memory^:=aux^;
     put(memory);
     aux:=aux^.next
    END;
   close(memory,lock)
  END;
 EXIT(program)
END;



FUNCTION badgame:boolean;

VAR cont:boolean;aux:link;

BEGIN
 aux:=base1;
 cont:=true;
 WHILE (aux<>nil)AND(cont) DO
  IF gt(trace,aux^.move) THEN aux:=aux^.next
  ELSE cont:=false;
 IF aux=nil THEN badgame:=false
 ELSE IF aux^.move=trace THEN badgame:=true
 ELSE badgame:=false


end;



PROCEDURE learn;

LABEL 10,20;
VAR
 aux,aux1,aux2:link;
 trace1:gametrack;
 ymax:ARRAY[0..7]OF integer;
 c,i,j,k:integer;
 cont,reduced:boolean;

BEGIN
 i:=40;learned:=true;
 WHILE trace[i]=0 DO i:=i-1;
 trace[i]:=0;trace[i-1]:=0;
 aux:=base;
 IF aux=nil THEN GOTO 10;
 IF gt(aux^.move,trace) THEN GOTO 10;
 cont:=true;
 WHILE (cont AND (aux<>nil)) do
  IF gt(trace,aux^.move) THEN
   BEGIN
```

Seguito figura 2.

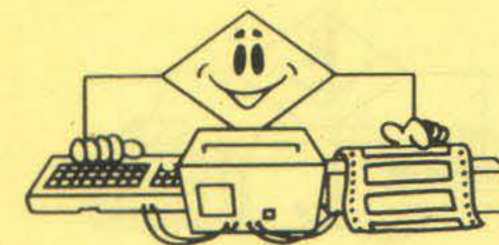
```
    aux1:=aux;
    aux:=aux^.next
   END
  ELSE cont:=false;
 new(aux2);
 WITH aux2^ DO
  BEGIN
   next:=aux;
   move:=trace;
   first:=starter
  END;
 aux1^.next:=aux2;
 GOTO 20;
 10: new(base);
 WITH base^ DO
  BEGIN
   next:=aux;
   move:=trace;
   first:=starter
  END;
 20:FOR j:=1 TO 40 DO
  IF trace[j]=0 THEN trace1[j]:=7
  ELSE trace1[j]:=trace[j];
  reduced:=false;
  REPEAT
   i:=i-2;
   trace[i]:=0;trace[i-1]:=0;
   trace1[i]:=7;trace1[i-1]:=7;
   aux:=base;j:=0;k:=0;
   WHILE gt(trace,aux^.move) DO aux:=aux^.next;aux1:=aux;
   WHILE (aux<>nil)AND(k<8) DO
    BEGIN
     k:=k+1;
     IF not(gt(aux^.move,trace1)) THEN
      BEGIN
        j:=j+1;
        aux:=aux^.next
       END
     END;
    FOR k:=0 TO 7 DO ymax[k]:=0;
    FOR k:=1 TO 40 DO ymax[trace[k]]:=ymax[trace[k]]+1;
    c:=0;FOR k:=1 TO 7 DO IF ymax[k]<6 THEN c:=c+1;
    IF j>=c-numcol[i DIV 2] THEN WITH aux1^ DO
     BEGIN
      next:=aux;
      move:=trace;
      first:=starter
     END
    ELSE reduced:=true;
   UNTIL reduced OR (i<2)
END;




PROCEDURE grmove(who,column:integer);

VAR i,c:integer;

BEGIN
 i:=0;c:=3+26*(column-1);
 IF who>0 THEN who:=0 ELSE who:=24;
 REPEAT
  drawblock(aux,6,24,0,24,24,c,9+26*(6-i),8);
  drawblock(apple,6,who,0,24,24,c,9+26*(5-i),14);
  i:=i+1
 UNTIL board[i,column]<>0
END;



PROCEDURE domove(who,column:integer;VAR board:matrix);

VAR
 row:integer;
```

Seguito figura 2.

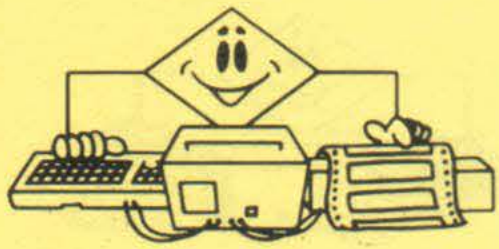
```
BEGIN
 row:=0;
 WHILE board[row+1,column]=emptydot DO row:=row+1;
 board[row,column]:=who
END;



FUNCTION value (column,whose:integer;board:matrix):integer;EXTERNAL;



FUNCTION rvalue(column,depth,whose,dummy:integer;board:matrix):integer;

VAR
 i,k,bestvalue:integer;

BEGIN
 k:=value(column,whose,board);
 bestvalue:=abs(k);

IF (bestvalue=fourline)OR(bestvalue=-illegal)OR(depth=maxdepth)THEN
 rvalue:=k-depth*whose
ELSE
 BEGIN
  domove(compudot*whose,column,board);
  IF whose=1 THEN
   BEGIN
    bestvalue:=maxint;
    FOR i:=1 TO 7 DO
     BEGIN
      k:=rvalue(i,depth+1,-whose,bestvalue,board);
      IF k<bestvalue THEN
       BEGIN
        bestvalue:=k;
        IF bestvalue<=dummy THEN
         BEGIN
          rvalue:=bestvalue;
          EXIT(rvalue)
         END
       END
     END;
    rvalue:=bestvalue
   END
  ELSE
   BEGIN
    bestvalue:=-maxint;
    FOR i:=1 TO 7 DO
     BEGIN
      k:=rvalue(i,depth+1,-whose,bestvalue,board);
      IF k>bestvalue THEN
       BEGIN
        bestvalue:=k;
        IF bestvalue>=dummy THEN
         BEGIN
          rvalue:=bestvalue;
          EXIT(rvalue)
         END
       END
     END;
    rvalue:=bestvalue
   END
 END
END;



FUNCTION choice : integer;

VAR
 i,k,bestval,bestmove:integer;
 chance:boolean; j:0..7;
```

Seguito figura 2.

```
BEGIN
 chance:=false;
 numcol[movenum DIV 2]:=0;
 bestval:=-maxint;j:=0;
 FOR i:=1 TO 7 DO IF board[1,i]=0 THEN j:=j+1;
 CASE j OF
  1:maxdepth:=1;
  2:maxdepth:=9;
  3:maxdepth:=7;
  4:maxdepth:=5;
  5,6,7:maxdepth:=4
 END;
 IF maxdepth>(43-movenum) THEN maxdepth:=43-movenum;
 FOR i:=1 TO 7 DO
  BEGIN
   time:=24-time;
   drawblock(apple,6,0,0,24,24,186,95,4);
   drawblock(apple,6,time,0,24,24,186,95,14);
   k:=rvalue(i,1,1,-maxint,board);
   trace[movenum]:=abs(mirror-i);
   IF k>-490 THEN IF badgame THEN k:=-490;
   IF k>-490 THEN chance:=true;
   IF k<-490 THEN numcol[movenum DIV 2]:=numcol[movenum DIV 2]+1;
   IF k>bestval THEN
    BEGIN
     bestval:=k;
     bestmove:=i
    END
  END;
 choice:=bestmove;
 IF NOT(chance) AND (status=tracing) THEN
 BEGIN
  status:=learning;
  IF memavail>1024 THEN learn
 END
END;



FUNCTION input(c1:integer):integer;

VAR ch:char;n:integer;



BEGIN
 IF keypress THEN read(keyin);
 n:=c1-1;
 REPEAT
  read(ch);
  drawblock(apple,6,24,0,24,24,3+26*n,165,0);
  CASE ch OF
   ',':BEGIN n:=(n-1);IF n<0 THEN n:=6 END;
   '.':n:=(n+1) MOD 7;
   ' ':IF board[1,n+1]=0 THEN
        BEGIN
         input:=n+1;
         EXIT(input)
        END;
   'A','a':BEGIN(*'Q','q':BEGIN*)
            quit:=true;
            EXIT(play)
           END;
   'I','i':BEGIN(*'T','t':BEGIN*)
            chartype(10);pencolor(none);
            moveto(192,38);wstring('Era davvero');(*wstring('Was my move');*)
            moveto(192,30);wstring('una cattiva');(*wstring('really bad?');*)
            moveto(192,22);wstring('mossa ? S/N');(*wstring('( Yes/No  )');*)
            read(keyin);
            IF ((keyin='S')OR(keyin='s'))AND
            (*IF ((keyin='Y')OR(keyin='y'))AND*)
```

APPLE

Seguito figura 2.

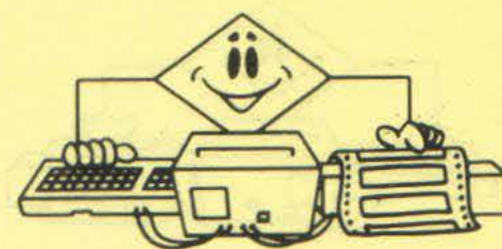
```
              (status=tracing)AND(movenum<39) THEN
               BEGIN
                status:=learning;
                trace[movenum+1]:=4;trace[movenum+2]:=4;
                IF memavail>1024 THEN learn
               END;
              home
             END;
    'R','r':EXIT(play);
    'S','s':sound:=NOT sound
   END;
   drawblock(apple,6,24,0,24,24,3+26*n,165,10)
  UNTIL false
END;



PROCEDURE draw;

VAR i:integer;

BEGIN
 FOR i:=1 TO 7 DO
  IF board[1,i]=0 THEN EXIT(draw);
 pencolor(none);moveto(192,30);chartype(10);
 wstring('...PATTA...');(*wstring('It''s a draw');*)
 read(keyin);home;EXIT(play)
END;



PROCEDURE four(column:integer);

VAR
 row,who,dx,dy,x,y,num:integer;

BEGIN
 row:=1;
 WHILE board[row,column]=0 DO row:=row+1;
 who:=board[row,column];
 FOR dx:=0 TO 1 DO
 FOR dy:=-1 TO 1 DO
 IF (dx<>0) OR (dy>0) THEN
  BEGIN
   x:=column;y:=row;num:=-1;
   REPEAT
    x:=x+dx;y:=y+dy;
    num:=num+1
   UNTIL board[y,x]<>who;
   x:=column;y:=row;
   REPEAT
    x:=x-dx;y:=y-dy;
    num:=num+1
   UNTIL board[y,x]<>who;
   IF num>3 THEN
    BEGIN
     IF who>0 THEN
      BEGIN
       pencolor(none);moveto(192,30);
       chartype(10);wstring('HO VINTO !!');(*wstring('wow I win !');*)
       sound2;read(keyin);home;EXIT(play)
      END


     ELSE
      BEGIN
       pencolor(none);moveto(192,30);
       chartype(10);wstring('HO PERSO!@*');(*wstring('ops, I lose');*)
       sound1;read(keyin);home;EXIT(play)
      END
    END
  END
END;
```

APPLE

Seguito figura 2.

```
PROCEDURE play ;

VAR
 num,c1 : integer;
 cont:boolean;

BEGIN
 c1:=i;base1:=base;
 REPEAT
  num:=input(c1);
  IF (notset AND (num<>4)) THEN
   BEGIN
    notset:=false;
    IF num>4 THEN mirror:=8
   END;
  IF status=tracing THEN
   BEGIN
    movenum:=movenum+1;
    trace[movenum]:=abs(mirror-num);
    IF movenum=40 THEN status:=learning
   END;
  domove(humandot,num,board);
  grmove(-1,num);
  four(num);draw;
  movenum:=movenum+1;
  c1:=choice;
  IF (notset AND (c1<>4)) THEN
   BEGIN
    notset:=false;
    IF c1>4 THEN mirror:=8
   END;
  IF status=tracing THEN
   BEGIN
    trace[movenum]:=abs(mirror-c1);
    IF movenum=40 THEN status:=learning
   END;
  domove(compudot,c1,board);
  grmove(1,c1);
  drawblock(apple,6,24,0,24,24,3+26*(c1-1),165,10);
  four(c1);
  cont:=true;
  WHILE (base1<>nil)AND cont DO
   IF gt(trace,base1^.move) THEN base1:=base1^.next
   ELSE cont:=false;
  draw;sound0
 UNTIL FALSE
END;

BEGIN
 time:=0;sound:=true;quit:=false;
 remember;start;
 status:=learning;learned:=false;
 REPEAT
  FOR i:=1 to 7 DO
  FOR j:=1 to 6 DO
  board[j,i]:=emptydot;
  FOR i:=0 to 7 DO
   BEGIN
    board[i,0]:=outdot;
    board[i,8]:=outdot
   END;
  FOR i:=0 TO 8 DO
   BEGIN
    board[0,i]:=outdot;
    board[7,i]:=outdot
   END;
  chartype(10);pencolor(none);
  moveto(192,38);wstring('Chi   muove');(*wstring('Who  of  us');*)
  moveto(192,30);wstring('per primo ?');(*wstring('goes first?');*)
  moveto(192,22);wstring('(Comp/Util)');(*wstring('(Comp/User)');*)
  notset:=true;mirror:=0;
  IF status=learning THEN status:=tracing;
  FOR i:=1 TO 40 DO trace[i]:=0;
  READ (keyin);home;randomize;
```

APPLE

Seguito figura 2.

```
   IF (keyin='A') OR (keyin='a') THEN memorize;
   (*IF (keyin='Q') OR (keyin='q') THEN memorize;*)
   IF (keyin='C') OR (keyin='c') THEN
    BEGIN
     base:=basec;starter:=FC;movenum:=1;
     i:=random MOD 3 + 3;
     IF i<>4 THEN BEGIN notset:=false;IF i>4 THEN mirror:=8 END;
     trace[1]:=abs(mirror-i);
     domove(compudot,i,board);
     grmove(1,i);
     sound0;
     drawblock(apple,6,24,0,24,24,3+26*(i-1),165,10);
     play;
     basec:=base
    END
   ELSE
    BEGIN
     base:=baseh;starter:=FH;
     movenum:=0;
     drawblock(apple,6,24,0,24,24,81,165,10);
     i:=4;play;
     baseh:=base
    END;
   IF quit THEN memorize;
   start1
  UNTIL false
END.
```

```
        .FUNC VALUE,3
;FUNCTION value(column,whose:integer;board:matrix):integer;

RETURN  .EQU 00
BOARD   .EQU 02
BASE    .EQU 04
FRIEND  .EQU 05
DIR     .EQU 06
DIST    .EQU 07
MYDOTS  .EQU 08
FREE    .EQU 09
STEP    .EQU 0A
VAL     .EQU 0B
SCORE   .EQU 0D
INLINE  .EQU 0E

;prendi i dati dallo stack
        PLA                 ;salva l'indirizzo di ritorno
        STA RETURN
        PLA
        STA RETURN+1
        PLA                 ;scarta i soliti 4 byte inutili
        PLA
        PLA
        PLA
        PLA                 ;salva l'indirizzo del tabellone di gioco
        STA BOARD
        PLA
        STA BOARD+1
        PLA                 ;calcola il valore assegnato alla pedina
        PLA                 ;del giocatore per cui calcoli la
        BMI $1              ;funzione value
        LDA #5
        BPL $2
     $1 LDA #0FB
     $2 STA FRIEND
        PLA                 ;calcola l'indirizzo relativo di
        ASL A               ;board[0,column]; il carry resta resettato
        TAY                 ;calcola l'indirizzo relativo dell'ultima
     $3 STY BASE            ;casella libera della colonna partendo
        TYA                 ;dall'alto e mettilo in BASE
        ADC #18.
        TAY
        LDA @BOARD,Y
        BEQ $3
        PLA
;calcola il valore della colonna
```

Seguito figura 2.

```
         CPY #36.         ;controlla se la mossa e' valida
         BCS LEGAL        ;se no poni VAL=-10000
         LDA #240.
         STA VAL
         LDA #216.
         STA VAL+1
         JMP END
LEGAL    LDA #0           ;poni le condizioni iniziali
         STA VAL
         STA VAL+1
         LDA #3
         STA DIR
SCANNER  LDY DIR          ;calcola SCORE per ogni direzione
         LDA POSINC,Y     ;e nel caso aggiungilo a VAL; se ci
         STA STEP         ;sono quattro pedine in fila esci
         LDA #0           ;dalla subroutine con VALUE=500
         STA MYDOTS
         STA FREE
         STA SCORE
         JSR SCAN
         LDY DIR
         LDA NEGINC,Y
         STA STEP
         JSR SCAN
         LDA FREE         ;aggiungi SCORE a VAL solo
         CMP #3           ;se ci sono almeno tre spazi liberi
         BCC $1           ;o occupati da pedine dello stesso
         CLC              ;giocatore di fianco all'elemento base
         LDA SCORE
         ADC VAL
```

APPLE

Seguito figura 2.

```
         STA VAL
         LDA #0
         ADC VAL+1
         STA VAL+1
         LDA MYDOTS      ;controlla se ci sono almeno
         CMP #3          ;quattro pedine in fila
         BCC $1
         LDA #244.       ;nel caso esci dalla subroutine
         STA VAL
         LDA #1.
         STA VAL+1
         JMP END
      $1 DEC DIR
         BPL SCANNER
END      LDA FRIEND      ;poni il segno opportuno a VAL
         BPL END1
         SEC
         LDA #0
         SBC VAL
         STA VAL
         LDA #0
         SBC VAL+1
         STA VAL+1
END1     LDA VAL+1
         PHA
         LDA VAL
         PHA
         LDA RETURN+1
         PHA
         LDA RETURN
         PHA
         RTS

;subroutine per l'analisi delle singole direzioni

SCAN     LDA FRIEND      ;poni le condizioni iniziali
         STA INLINE
         LDA #10.
         LDY BASE        ;il registro Y contiene l'indirizzo
SCLOOP   TYA             ;relativo per l'elemento da esaminare
         CLC
         ADC STEP
         TAY
         LDA @BOARD,Y    ;esamina il sihgolo elemento; se
         TAX             ;dall'elememto iniziale a questo incluso
         EOR INLINE      ;tutte le caselle contengono pedine del
         BEQ $1          ;giocatore per cui calcoli VALUE incrementa MYDOTS
         LDA #0F         ;un valore qualunque non usato nel tabellone
         STA INLINE
         BNE $2
      $1 INC MYDOTS
      $2 LDA SCORE
         CPX #00
         BEQ $4
         CPX FRIEND
         BEQ $3
         RTS             ;se hai incontrato il bordo del tabellone
      $3 CLC             ;o una pedina avversaria esci dalla subroutine
         ADC #5
      $4 CLC
         ADC DIST
         STA SCORE
         DEC DIST
         INC FREE
         LDA DIST
         CMP #7.
         BNE SCLOOP
         RTS


POSINC   .BYTE 18.+2.,2.,2.-18.,-18.
NEGINC   .BYTE -18.-2.,-2.,18.-2.,18.

         .END
```

Chi ti può offrire questo computer *per 750.000 Lire??

!!! COMPUMAIL PUO' !!!

Compumail: un'organizzazione ad esclusivo servizio dell'utente finale per corrispondenza, naturalmente

Desidero ricevere il materiale sotto indicato:

QUANTITA'	DESCRIZIONE	PREZZO UNITARIO
* n.	BEAP computer, scatolato come da foto, 48 K/CPU 6502, tastiera M/m con tasti funzione, firmware su EPROM 2732, interamente zoccolato - alimentatore switching	750.000
n.	Disk Driver 5 1/4"	470.000
n.	Monitor PHILIPS PCT 1202 antirif. fosfori verdi	250.000
n.	Monitor PHILIPS PCT 1202 antirif. fosfori arancio	265.000
n.	Stampante STAR DP 510 80 col/100 cps F/T	750.000
n.	Stampante STAR GEMINI-10X 80 col/120 cps F/T	900.000
n.	Stampante STAR GEMINI-15X 132 col/120 cps F/T	1.270.000
n.	Interfaccia parallela per stampante	70.000
n.	Interfaccia seriale RS 232	95.000
n.	Scheda Disk Controller	85.000
n.	Scheda espansione 16 K	95.000
n.	Scheda CPU Z80	130.000
n.	Joystick	35.000
n.	Paddles	30.000
n.	Scheda 80 colonne con soft/switch	180.000
n.	Modulatore VHF con switch antenna/computer	35.000
n.	Alim. switching: 5V 5A/12V 2,5A/-12V 0,5A/-5V 0,5A; scatolato con filtro rete, presa e interr. + cordone 220V	140.000

BEAP MONITORS E STAMPANTI SONO COMPLETI DI MANUALE D'USO E CORDONE DI ALIMENTAZIONE A NORME

SPECIALE

n.	SISTEMA "BEAP BASE" Composto da: 1 BEAP Computer. 1 Monitor Philips f. verdi. 1 Disk driver + controller	1.500.000
n.	SISTEMA "BEAP SUPER" Composto da: 1 sist. "BEAP BASE" + 1 Disk driver. 1 Stampante STAR DP 510 + Interfaccia	2.750.000

SPECIALE

CONTRIBUTO SPESE IMBALLO E SPEDIZIONE

consegna a mezzo pacco postale raccomandato c/assegno .... Lit. 15.000
consegna rapida a domicilio a mezzo corriere giornaliero (TRACO, etc.) ...... Lit. 25.000

Totale imponibile Lit. ... .
Più IVA 18% Lit ... .

IMPORTO TOTALE Lit ... .

che vorrete inviarmi, assieme alla fattura, al seguente indirizzo:

Nome .... Cognome ....

Ragione Sociale (se Ditta) ....

Via .... CAP .... Città .... Prov. ....

Codice Fiscale .... P. IVA (se Ditta) ....

Numero telefonico ....

MODALITA':

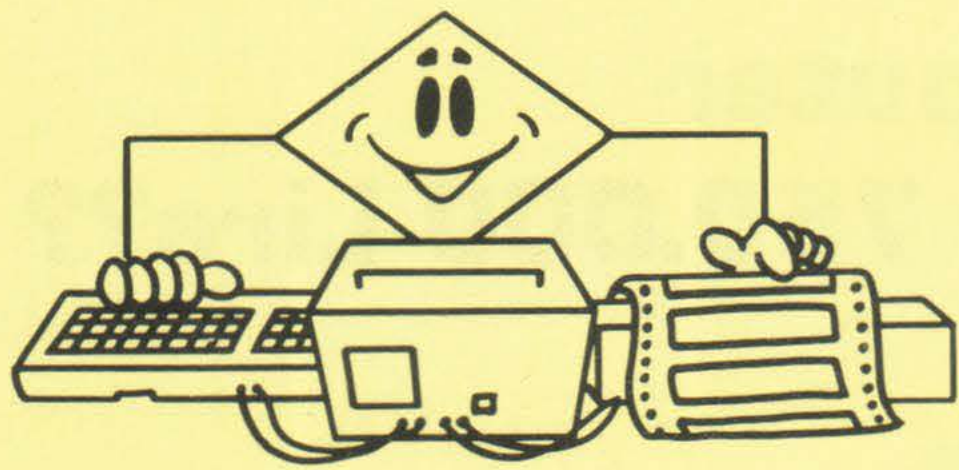
(*) Allego anticipatamente assegno di Lit. .... (come da riquadro) quale contributo alle spese di spedizione, impegnandomi a saldare contrassegno l'importo totale di Lit. .... come da calcolo sopra effettuato.

(*) Allego anticipatamente assegno per l'importo totale di Lit. .... come da calcolo sopra effettuato; restano a vostro carico le spese di spedizione. Qualora il conteggio da me effettuato dovesse risultare errato mi rimborserete a stretto giro di posta l'eventuale differenza pagatavi in più o mi addebiterete in contrassegno l'importo mancante.

DATA ..../............../.............. FIRMA ....

(*) Desidero ricevere dettagliate informazioni sui seguenti prodotti:

( )Computers ( ) Stampanti ( ) Disk drivers ( ) Monitors ( ) Espansioni ( ) Interfacce ( ) Software

Ritagliare e spedire a:
COMPUMAIL - Casella Postale 404 - 13051 Biella (VC)

Disegno assistito dal computer

Questo programma è stato realizzato dopo aver osservato, nel corso di varie mostre ed esposizioni, quello che possono fare i computer quando vengono usati nella cosiddetta "computer assisted design" cioè nella progettazione assistita dal computer o nella "computer assisted drawing" cioè nel disegno assistito dal computer.

di **M. Cerofolini**

Figura 1 - Shape studiate per il disegno di circuiti elettrici.

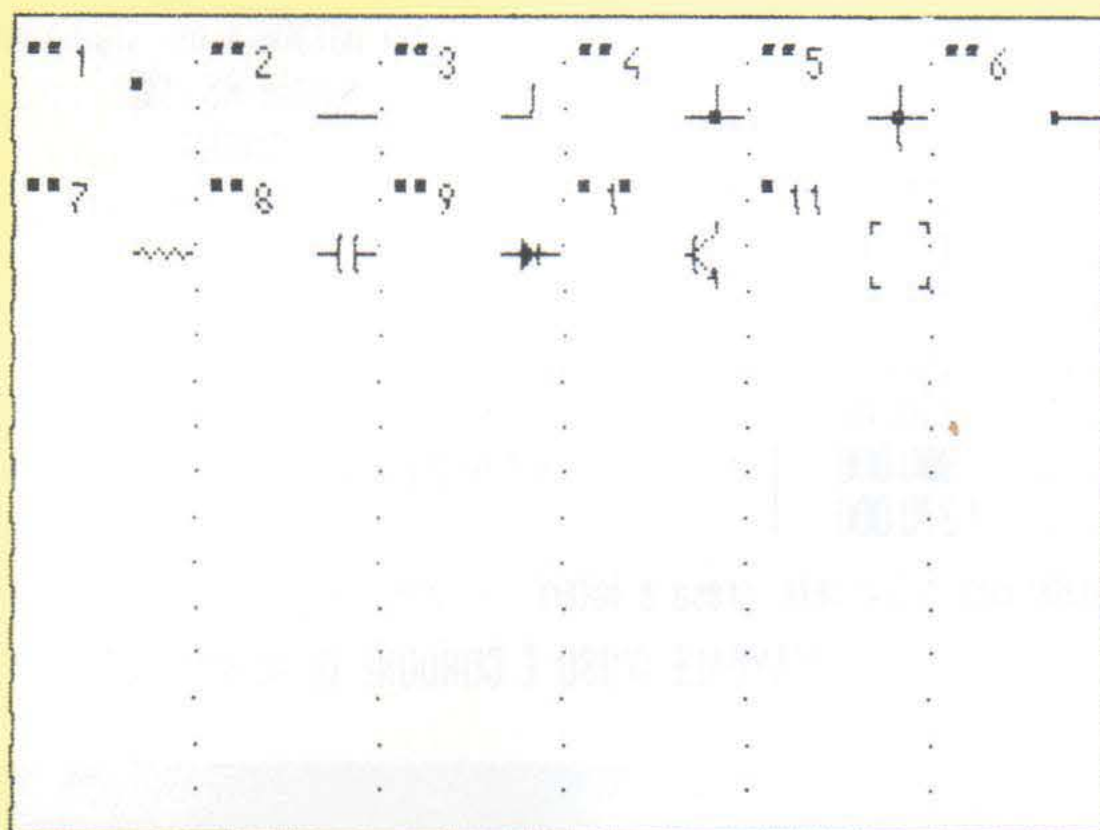



Figura 2 - Disegno ottenuto con l'insieme di più shape.

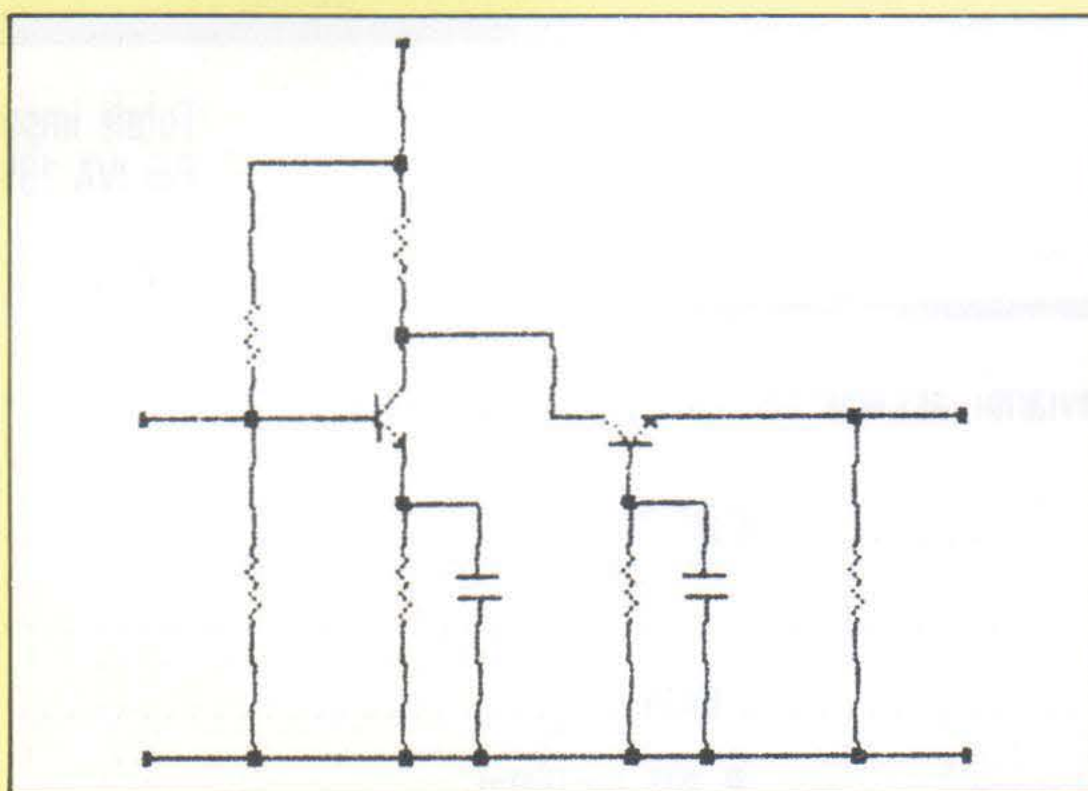
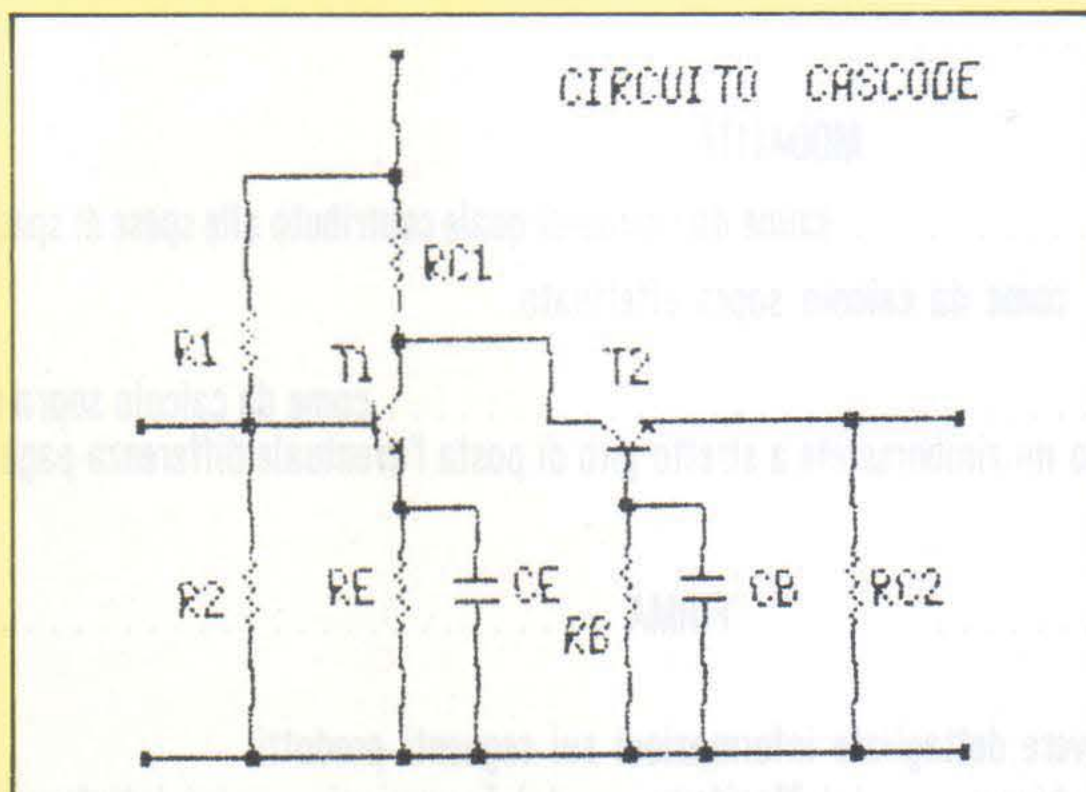



Figura 3 - Stesso disegno di figura 2 completato dalle scritte ottenute dal programma presentato.

Guidati da opportuni programmi e corredati da accessori quali plotter e tavolette grafiche, i computer sono in grado di realizzare progetti di straordinaria complessità come ad esempio circuiti stampati, mappe e carte geografiche, schemi elettrici, schemi di montaggio per wire-wrap ecc. ecc.

Ma cosa può fare l'utente di un personal computer dotato di capacità grafiche limitate e che non dispone di tavolette grafiche e/o plotter? Con l'ausilio dei programmi che seguono e con l'ormai diffusissimo Apple II (che dispone di una grafica di buona risoluzione) e di una stampante grafica, si possono ottenere dei risultati che fino a pochi anni fa sembravano alla portata solo di computer del costo di centinaia di milioni.

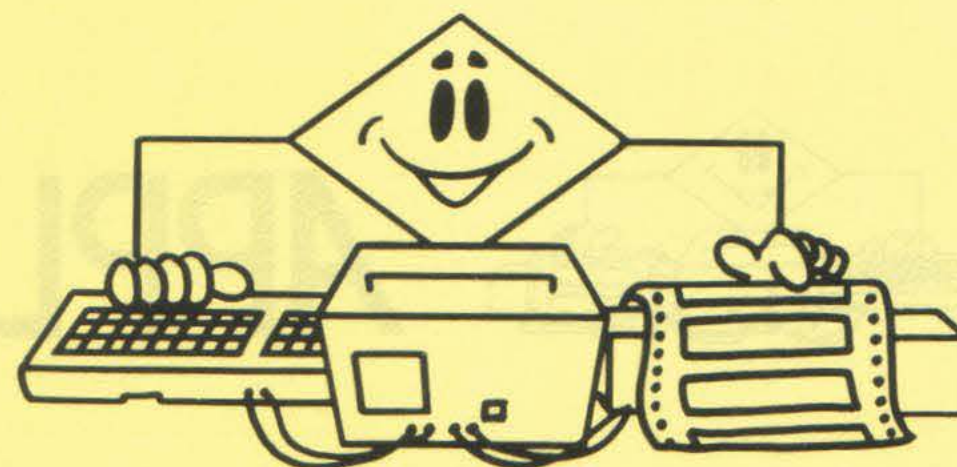
La procedura di "disegno assistito dal computer" (o computer aided design CAD) che verrà di seguito descritta è costituita da un programma per disegnare, da un programma per aggiungere un testo alfanumerico al disegno preparato in precedenza e da una serie di disegni elementari (shape table) che costituiscono i mattoni con i quali poi verrà realizzato un disegno più complesso.

Il programma per disegnare

Il programma per disegnare si basa sulla possibilità dell'Apple II di utilizzare la "shape table". Questa non è altro che una tabella di disegni già predisposti che possono essere preparati in precedenza e memorizzati su disco. Dopo aver richiamato dalla "mass storage" (cioè dall'unità a disco) la "shape table", i disegni che la costituiscono possono essere spostati in un punto qualsiasi della pagina grafica, possono venire ruotati, ingranditi e presentati in vari colori. Se opportunamente studiate nella loro forma, le shape possono essere combinate insieme per formare disegni anche molto complessi. La figura 1 presenta le shape che sono state studiate per il disegno di circuiti elettrici. Sono presenti i simboli più comuni usati negli schemi elettrici e le linee, le intersezioni e gli angoli.

Il programma presentato nel listing 1 divide la pagina grafica in una griglia di 18 quadrati per 10.

Ciascuno di questi quadrati costituisce la sede di una delle shape di 15 x 15 punti descritte in pre-

cedenza. Il programma permette poi, attraverso opportuni comandi, di selezionare la shape opportuna, di spostarla in un qualsiasi punto dello schermo, di ruotarla (con incrementi di 90 gradi) e di cancellarla dallo schermo.
Tutte le operazioni descritte vengono visualizzate sulla pagina grafica e contemporaneamente memorizzate in due opportune matrici (D% e R%).

I comandi che permettono di manipolare le shape vengono dati con la pressione di un singolo tasto e sono visibili in figura 4.
Il cursore è costituito dalla shape numero 11 (figura 1) e si presenta sotto forma di quattro angoli lampeggianti. Essi delimitano la shape e servono ad indicare in quale punto dello schermo grafico verranno effettuate le operazioni descritte in precedenza.

È interessante soffermarci un attimo sulla routine che inizia alla linea 4410. Questa permette di accettare un carattere da tastiera mentre il programma BASIC deve effettuare un'altra operazione. In questo programma era necessario fare lampeggiare in continuazione il cursore e nello stesso tempo accettare un comando da tastiera. Usando un normale statement GET ciò non sarebbe stato possibile perchè, nel momento in cui questo statement viene eseguito, non cede il controllo fino a che un carattere non viene battuto sulla tastiera.

Come già accennato precedentemente la figura 1 riporta le shape corrispondenti ai numeri da 1 a 9. Digitando uno di questi numeri mentre il cursore lampeggia la corrispondente shape appare sotto il cursore cancellando l'eventuale disegno che vi era in precedenza. I comandi di rotazione e cancellazione visti in precedenza agiscono sulla shape che è posta sotto il cursore. Mettendo insieme varie shape si può costruire un disegno come quello indicato in figura 2. Occorre inoltre notare che le shape di figura 1 sono presentate nelle loro orientazioni di base ma nel disegno esse possono essere ruotate a piacimento.

Quando il disegno è completo si preme il tasto CTRL-F che esce dal loop di accettazione comandi e chiede sotto quale nome vada salvato il disegno. Il programma salva su disco l'area di memoria corrispondente alla pagina grafica che contiene il disegno ($8000 per lunghezza $8000).

1-9	Scelgono una delle shape a disposizione (massimo 9).
I,J,K,M	Muovono il cursore nelle 4 direzioni.
C	Cancella la shape nella posizione del cursore.
<	Ruota la shape in senso antiorario.
>	Ruota la shape in senso orario.
CTRL F	Esce dal programma di disegno.

Figura 4 - Comandi che permettono di manipolare le shape.

```
7530-  3D 00 02 01 08 01 13 01
7538-  19 01 1E 01 27 01 38 01
7540-  46 01 4E 01 5C 01 63 01
7548-  6B 01 72 01 7C 01 85 01
7550-  88 01 8E 01 90 01 96 01
7558-  A1 01 A8 01 B2 01 BB 01
7560-  C5 01 CE 01 D9 01 E0 01
7568-  EC 01 F5 01 FA 01 FE 01
7570-  06 02 0E 02 14 02 1B 02
7578-  28 02 33 02 3E 02 48 02
7580-  52 02 5C 02 64 02 70 02
7588-  7B 02 83 02 8C 02 9A 02
7590-  A2 02 AE 02 BA 02 C3 02
7598-  CB 02 D5 02 E0 02 EA 02
75A0-  F2 02 FB 02 07 03 14 03
75A8-  1F 03 28 03 34 03 00 00
```

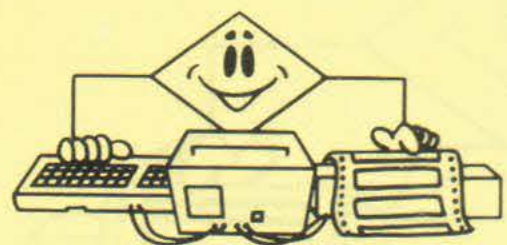
Figura 5 - Shape table dell'alfabeto ASCII.

```
*
7530-  0B 00 1A 00 20 00 2B 00
7538-  36 00 4C 00 69 00 77 00
7540-  88 00 9E 00 BA 00 DE 00
7548-  F8 00 3E 24 2D 36 04 00
7550-  92 92 2A 2D 2D 2D 2D 2D
7558-  2D 25 00 92 92 2A 2D 2D
7560-  2D 24 24 24 24 00 92 92
7568-  2A 2D 2D 2D 2D 2D 2D E5
7570-  DB 1B 3F 16 2D 0C 18 18
7578-  24 24 24 00 92 92 2A 2D
7580-  2D 2D 2D 2D 2D 25 08 18
7588-  08 18 08 D8 DB DB 36 36
7590-  36 6F 16 3F 0E 36 36 26
7598-  00 92 92 22 35 3E 4C 2D
75A0-  2D 2D 2D 2D 2D 04 00 92
75A8-  92 2A 0C 0E 0E 0C 0C 0E
75B0-  0E 0C 0C 0E 0E 0C 25 00
75B8-  92 92 2A 2D 2D 24 24 96
75C0-  12 36 6E 09 24 24 24 24
75C8-  95 12 2D 2D 04 00 92 92
75D0-  2A 2D 2D 24 B4 12 36 0D
75D8-  18 24 24 15 36 0D 18 2D
75E0-  FD 0B 24 96 2E 08 58 29
75E8-  25 0C 92 92 2A 25 24 B4
75F0-  92 36 26 08 18 08 60 0D
75F8-  18 0D 18 0D 18 24 96 92
7600-  DA 9B 15 15 15 15 0E 18
7608-  20 97 6F 32 04 00 2D DE
7610-  B6 92 92 12 36 2D 4D 49
7618-  49 29 25 24 08 18 08 18
7620-  08 18 08 18 20 3C 27 00
7628-  00 00 00 00 00 00 00 00
```

Figura 6 - Shape table dei simboli.

Programma di scrittura

Il programma di scrittura (listato 2) permette l'inserimento di tutti i caratteri del set ASCII orientati in qualsiasi modo nel disegno preparato e salvato in precedenza. Il principio di funzionamento è molto simile a quello precedentemente descritto per il programma di disegno. Quello che cambia sono le shape che questa volta sono le lettere del set ASCII. Per spostare il cursore senza scrivere niente ci si comporta come quando si usa lo screen editor della normale tastiera cioè facendo precedere i soli movimenti dal tasto escape. Il movimento corrisponde ad uno solo dei punti dello schermo ad alta risoluzione. Se si devono effettuare degli spostamenti più ampi si può aumentare il passo premendo uno qualsiasi dei numeri mentre si è nella funzione di movimento cursore. Anche con questo programma le scritte possono essere orientate in una qualsiasi delle quattro possibili direzioni.

Per iniziare a scrivere in una direzione determinata è sufficiente sceglierla mediante i tasti <E>. Da quel momento tutti i caratteri scritti avranno l'orientamento prescelto. Per tornare indietro si può usare la freccia a sinistra mentre la freccia a

destra non è abilitata. Qualsiasi comando errato non viene comunque accettato e viene emessa una segnalazione acustica.

La figura 3 mostra il disegno di figura 2 completato delle scritte ottenute con questo programma.

Stampa dei disegni

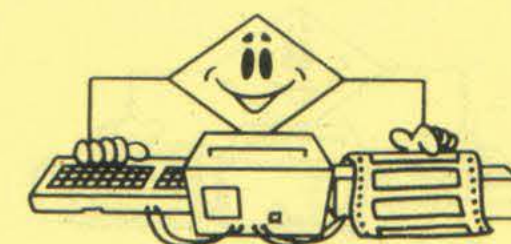
La stampa dei disegni così ottenuti è permessa su una qualsiasi delle stampanti grafiche che sono disponibili per l'Apple II. Io ho usato la stampante grafica Itoh messami gentilmente a disposizione dalla ditta "Audio Canalgrande Informatica" di Modena che mi ha anche fornito il programma di stampa della pagina grafica. Il procedimento per stampare il disegno è molto semplice e consiste nel caricare il disegno da dischetto e lanciare quindi il programma di stampa contenuto nell'interfaccia grafica. Questo programma pone la stampante in modo grafico e poi invia tutti i punti della pagina grafica alla stampante che ne stampa 7 alla volta in verticale.

Ampliamenti

Al programma presentato sono apportabili numerose modifiche ed ampliamenti.
La modifica più interessante è, a mio avviso, quella di considerare la pagina grafica come una finestra su di un disegno molto più grande.
Un'altra possibilità è data dal fatto che il disegno finale può essere fatto tramite un plotter invece che con una stampante grafica. All'interno del programma si dispone infatti della matrice D% che contiene tutte le shape del disegno mentre la matrice R% contiene l'orientamento delle varie figure. Con un opportuno programma è quindi possibile riportare su plotter il disegno memorizzato nelle due matrici indicate.

Naturalmente le shape presentate sono relative a disegni di schemi elettrici, ma possono essere cambiate a piacere per eseguire disegni di tipo totalmente diverso. Se volete avere degli spunti su quali possono essere i disegni elementari, ovvero le shape table utili ad una determinata applicazione vi consiglio di comprare, in una qualsiasi cartoleria, il catalogo Letraset e formare le shape copiando i disegni che vi sono riportati.

Listato 1 - Programma atto alla divisione della pagina grafica in una griglia di 18x10.

```
100  HGR : HOME : HCOLOR= 3: SCALE= 1: ROT= 0
320  REM ---- CARICAMENTO SHAPES TABLE -----
330  PRINT  CHR$ (4);"BLOAD EL1"
340 SH =  INT (30000 / 256):SL = 30000 - 256 * SH
350  POKE 232,SL: POKE 233,SH: REM  SET STPOINTER
360  HIMEM: 30000
366  SCALE= 1: ROT= 0
370  REM  ---- INIZIO PROGRAMMA -----------
380  DIM D%(18,10),R%(18,10)
381 R = 0:X = 1:Y = 1: GOSUB 6000:C = 11
382  POKE  - 16384,0
400  REM  ---- LOOP PRINCIPALE -----------
800  REM ----------------------------------
4400  REM ---- MOVIMENTI DEL CURSORE ---------
4410 T =  PEEK ( - 16384): IF T > 127 THEN  POKE  - 16384,0: GET T$: GOTO
     4500
4420  ROT= 0
4430  XDRAW C AT (X - 1) * 15 + TA,(Y - 1) * 15 + TA
4435  FOR I = 1 TO 200: NEXT
4440  XDRAW C AT (X - 1) * 15 + TA,(Y - 1) * 15 + TA
4441  FOR I = 1 TO 200: NEXT
4450  GOTO 4410
4500  REM ---------------------------------
4510  IF T$ = "I" THEN Y = Y - 1: GOTO 4545
4520  IF T$ = "M" THEN Y = Y + 1: GOTO 4545
4530  IF T$ = "J" THEN X = X - 1: GOTO 4545
4540  IF T$ = "K" THEN X = X + 1: GOTO 4545
4542  GOTO 4550
4545  IF X = 0 THEN X = 1
4546  IF X = 19 THEN X = 18
4547  IF Y = 0 THEN Y = 1
4548  IF Y = 11 THEN Y = 10
4549  GOSUB 6000: GOTO 4400
4550  REM  ----- ROTAZIONE ---------
4560  IF T$ = "<" OR T$ = ">" THEN 4580
4570  GOTO 4640
4580  IF D%(X,Y) = 0 THEN  PRINT  CHR$ (7): GOTO 4400
4590  GOSUB 8000: REM  CANCELLA IL PEZZO
4600  IF T$ = "<" THEN R%(X,Y) = R%(X,Y) - 16: IF R%(X,Y) =  - 16 THEN R%(
     X,Y) = 48
4610  IF T$ = ">" THEN R%(X,Y) = R%(X,Y) + 16: IF R%(X,Y) = 64 THEN R%(X,Y
     ) = 0
4620  GOSUB 9000: REM  DISEGNA IL PEZZO
4630  GOTO 4400
4640  REM  ---- CANCELLA IL PEZZO ---------
4650  IF T$ < > "C" THEN 4700
4660  IF D%(X,Y) = 0 THEN  PRINT  CHR$ (7): GOTO 4400
4670  GOSUB 8000: REM  CANCELLA IL PEZZO
4680 D%(X,Y) = 0
```

Seguito listato 1.

```
4690  GOTO 4400
4700  REM  ---- POSIZIONA IL PEZZO -----------
4705  IF  ASC (T$) = 6 THEN 20000
4710  GOSUB 9500
4720  IF ER = 1 THEN  PRINT  CHR$ (7): GOTO 4400
4730  IF D%(X,Y) <  > 0 THEN  GOSUB 8000
4740 D%(X,Y) = S: GOSUB 9000
4750  GOTO 4400
6000  REM  ---- TRASF. COORDINATE ----------
6005  IF R%(X,Y) = 0 THEN X2 = 0:Y2 = 0
6010  IF R%(X,Y) = 16 THEN X2 = 14:Y2 = 0
6020  IF R%(X,Y) = 32 THEN X2 = 14:Y2 = 14
6030  IF R%(X,Y) = 48 THEN X2 = 0:Y2 = 14
6040 X1 = (X - 1) * 15 + TA + X2
6050 Y1 = (Y - 1) * 15 + TA + Y2
6060  RETURN
8000  REM ------ CANCELLA IL PEZZO A X,Y ROT=R%(X,Y)---
8010  HCOLOR= 0: ROT= R%(X,Y): GOSUB 6000: DRAW D%(X,Y) AT X1,Y1
8020  RETURN
9000  REM  ----- DISEGNA IL PEZZO A X,Y ROT =R%(X,Y) ----
9010  HCOLOR= 3: ROT= R%(X,Y): GOSUB 6000: DRAW D%(X,Y) AT X1,Y1
9020  RETURN
9500  REM ---- RICERCA IN TABELLA DEL SIMBOLO ----
9510 ER = 0
9520  IF T$ < "1" OR T$ > "9" THEN ER = 1: GOTO 9600
9530 S =  VAL (T$) + 1
9600  RETURN
20000  REM  ----FINE PROGRAMMA -------------
20010  TEXT : HOME
20020  INPUT "NOME DISEGNO ? ";N$
20030  PRINT  CHR$ (4);"BSAVE ";N$;",A8192,L8192"
20040  END
```

APPLE

Listato 2 - Programma per l'inserimento di tutti i caratteri del set ASCII.

```
50  HGR : TEXT
100  HOME
320  REM  ----- CARICAMENTO SHAPES TABLE -----
330  PRINT  CHR$ (4);"BLOAD ALFABETO"
340 SH =  INT (30000 / 256):SL = 30000 - 256 * SH
350  POKE 232,SL: POKE 233,SH: REM  SET STPOINTER
360  HIMEM: 30000
366  SCALE= 1: ROT= 0: HCOLOR= 3
370  REM  ----- INIZIO PROGRAMMA -------------
380  DIM TA(30): REM  BUFER SCRITTURA
381 R = 0:X = 1:Y = 7:C = 3:DI = 5:ES = 0:I1 = 1:ST = 1
382  POKE  - 16384,0
400  REM  ----- CARICAMENTO DISEGNO ----------
407  HOME
410  INPUT "NOME DISEGNO ? ";N$
420  PRINT  CHR$ (4);"BLOAD ";N$
430  POKE  - 16304,0: POKE  - 16300,0: POKE  - 16297,0: POKE  - 16302,0
450  GOTO 4400
4400  REM  ----- MOVIMENTI DEL CURSORE ---------
4410 T =  PEEK ( - 16384): IF T > 127 THEN  POKE  - 16384,0: GET T$: GOTO
     4500
4420  ROT= R
4430  XDRAW C AT X,Y
4435  FOR I = 1 TO 200: NEXT
4440  XDRAW C AT X,Y
4441  FOR I = 1 TO 200: NEXT
4450  GOTO 4410
4500  REM  ------------------------------------------
4503  IF ES = 0 THEN 4800
4504  IF T$ > "0" AND T$ < ":" THEN ST =  VAL (T$): GOTO 4400
4505  IF T$ = "I" OR T$ = "M" OR T$ = "J" OR T$ = "K" OR T$ = "<" OR T$ =
     ">" THEN 4510
4506 ES = 0:C = 3:I1 = 1: GOTO 4400
4510  IF T$ = "I" THEN Y = Y - ST: GOTO 4545
4520  IF T$ = "M" THEN Y = Y + ST: GOTO 4545
4530  IF T$ = "J" THEN X = X - ST: GOTO 4545
4540  IF T$ = "K" THEN X = X + ST: GOTO 4545
4542  GOTO 4550
4545  IF X < 0 THEN X = 0
4546  IF X > 279 THEN X = 279
4547  IF Y < 0 THEN Y = 0
4548  IF Y > 159 THEN Y = 159
4549  GOTO 4400
4550  REM  ----- ROTAZIONE ----------
4560  IF T$ = "<" OR T$ = ">" THEN 4580
4570  GOTO 4640
4580  IF T$ = "<" THEN R = R - 16
4590  IF T$ = ">" THEN R = R + 16
4600  IF R =  - 16 THEN R = 48
4610  IF R = 64 THEN R = 0
4630  GOTO 4400
4640  REM  ----- SCRITTURA --------------------
4800  REM  ------------------------------------------
4810  IF  ASC (T$) = 27 THEN C = 2:ES = 1: GOTO 4400
4820  IF  ASC (T$) < > 8 THEN 4900
4830 I1 = I1 - 1
4840  IF I1 = 0 THEN  PRINT  CHR$ (7):C = 2: GOTO 4400
4850  IF R = 0 THEN X = X - DI
4855  IF R = 16 THEN Y = Y - DI
4856  IF R = 32 THEN X = X + DI
4858  IF R = 48 THEN Y = Y + DI
4860  XDRAW TA(I1) AT X,Y
4865  GOTO 4400
4900  IF  ASC (T$) = 6 THEN 7000: REM  SALVA IL DISEGNO
4903  IF  ASC (T$) < 33 OR  ASC (T$) > 90 THEN  PRINT  CHR$ (7): GOTO 4400
4905 LT =  ASC (T$) - 29
4910  XDRAW LT AT X,Y
4920 TA(I1) = LT:I1 = I1 + 1
4930  IF R = 0 THEN X = X + DI
4940  IF R = 16 THEN Y = Y + DI
4942  IF R = 32 THEN X = X - DI
4945  IF R = 48 THEN Y = Y - DI
4950  GOTO 4400
7000  REM  ----- SALVA IL DISEGNO -----------
7010  TEXT : HOME
7020  INPUT "NOME FILE ? ";A$
7030  PRINT  CHR$ (4);"BSAVE ";A$;",A8192,L8192"
```

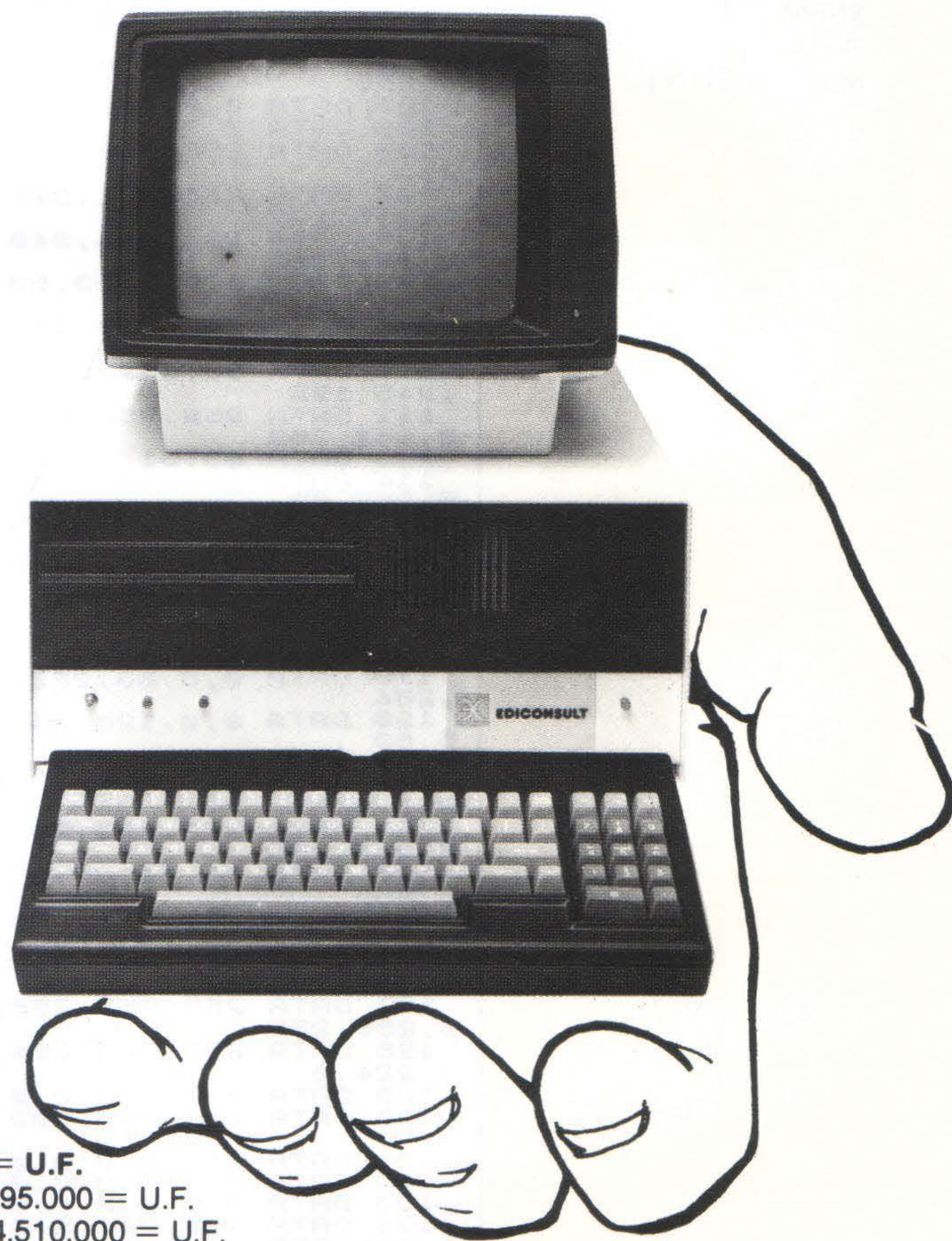
Novità

EDI MONO – EDI MULTI – EDI A RETE

(Prodotto in Italia!)

il miglior rapporto prezzo/prestazioni da EDICONSULT
ai prezzi più bassi del mercato

EDI 5"

Minifloppy da 5", da 400 a 1600 KB (da 1 a 4 unità: 400 KB)
Discofisso da 5", da 5 a 40 MB (da 1 a 4 unità: 5 o 10 MB)
Espansioni multiuser o a rete disponibili subito
Es. config.: 64 KB + 2FL5" (800 KB) + Video Term. = Lire 5.515.000 = U.F.
64 KB + 1FL5" (500 KB) + Disco 5" (5 MB) + Video Term. = Lire 8.445.000 = U.F.

EDI 8" e EDI 8" + 5"

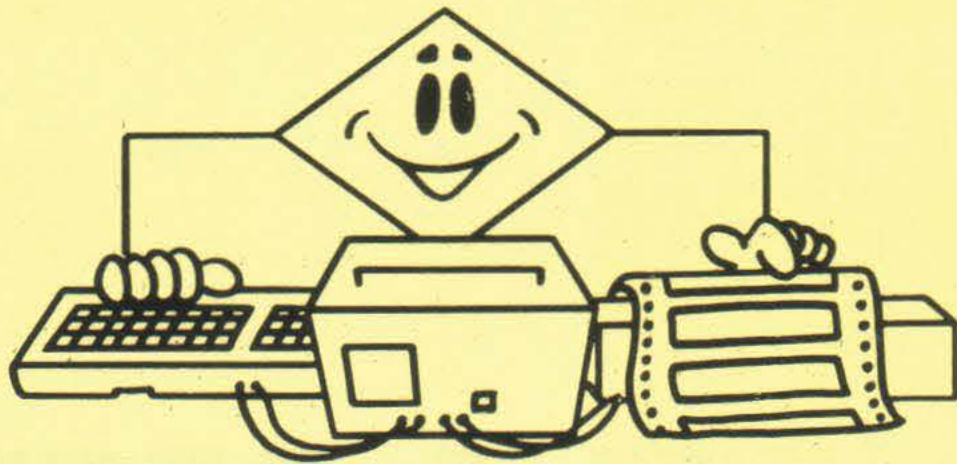
Floppy da 8", da 600 a 4800 KB (da 1 a 4 unità: 600 o 1200 KB)
Discofisso da 5", da 5 a 40 MB (da 1 a 4 unità: 5 o 10 MB)
Discofisso da 8", da 20 a 160 MB (da 1 a 4 unità: 20 o 40 MB)
Espansioni multiuser o a rete disponibili subito
Es. config.: 64 KB = 2FL8" (2.4MB) + Video Term. = Lire 6.680.000 = U.F.
64 KB + 1FL8" (1.2 MB) + Disco 5" (10 MB) + Video Term. = Lire 9.395.000 = U.F.
64 KB + 1FL 8" (1.2 MB) + Disco 8" (20 MB) + Video Term. = Lire 14.510.000 = U.F.

Abbreviazioni: FL = Floppy. KB = Kbytes. MB = Megabytes. U.F. = Utente Finale

Richiedete il prospetto di informazione tecnica a

EDICONSULT s.r.l.

Sede: 20052 MONZA - Via Rosmini, 3 - Telef. (039) 389.850-360.727

ricerchiamo concessionari per zone libere

ZX SPECTRUM

Doppietta per ZX. Due programmi a grafica predominante

Listato 1 - Programma per caricare e registrare schiera per SLOTMACH*: il listato si ottiene eseguendo il programma* LIST *allegato su cassetta che usa comandi* LPRINT.

```
  10 DIM x(5,136)
  20 FOR z=1 TO 5: FOR m=1 TO 13
6
  30 READ x(z,m)
  40 NEXT m: NEXT z
  50 SAVE "SLOTS" DATA x(): STOP
 101 DATA 0,0,0,0,0,0,1,3
 102 DATA 0,0,3,31,127,255,255,2
55
 103 DATA 0,0,192,248,254,255,25
5,255
 104 DATA 0,0,0,0,0,0,128,192
 105 DATA 7,7,15,15,31,31,63,63
 106 DATA 255,143,7,7,7,143,255,
254
 107 DATA 255,241,224,224,224,24
1,255,127
 108 DATA 224,224,240,240,248,24
8,252,252
 109 DATA 63,63,63,63,63,63,63,6
3
 110 DATA 252,249,243,231,207,15
9,175,3
 111 DATA 63,159,207,231,243,249
,245,192
 112 DATA 252,252,252,252,252,25
2,252,252
 113 DATA 63,31,31,15,7,3,1,0
 114 DATA 241,254,255,255,255,25
5,255,63
 115 DATA 143,127,255,255,255,25
5,255,252
 116 DATA 252,248,248,240,224,19
2,128,0
 117 DATA 255,143,31,31,31,143,2
55,255
 118 DATA 0,0,0,0,3,7,15,31
 119 DATA 0,0,63,255,255,255,255
,254
 120 DATA 0,0,128,240,252,254,25
5,63
 121 DATA 0,0,0,0,0,0,0,0
 122 DATA 31,63,127,127,255,255,
255,255
 123 DATA 252,252,252,254,255,25
5,255,255
 124 DATA 31,31,31,63,248,255,25
5,255
 125 DATA 192,248,255,0,0,255,24
8,192
 126 DATA 255,255,255,255,255,25
5,255,255
 127 DATA 255,255,255,255,255,25
5,255,255
 128 DATA 255,255,254,252,248,24
0,224,0
 129 DATA 0,0,0,0,0,0,0,0
 130 DATA 254,240,192,128,0,0,0,
0
 131 DATA 0,0,0,0,0,0,0,0
 132 DATA 0,0,0,0,0,0,0,0
 133 DATA 0,0,0,0,0,0,0,0
 134 DATA 255,241,248,248,248,24
1,255,255
 135 DATA 0,0,0,0,0,0,0,0
 136 DATA 0,0,1,15,63,127,255,25
2
 137 DATA 0,0,252,255,255,255,25
5,127
 138 DATA 0,0,0,0,192,224,240,24
8
 139 DATA 3,31,255,0,0,255,31,3
 140 DATA 248,248,248,252,31,255
,255,255
 141 DATA 63,63,63,127,255,255,2
55,255
 142 DATA 248,252,254,254,255,25
5,255,255
 143 DATA 0,0,0,0,0,0,0,0
 144 DATA 255,255,127,63,31,15,7
,0
```

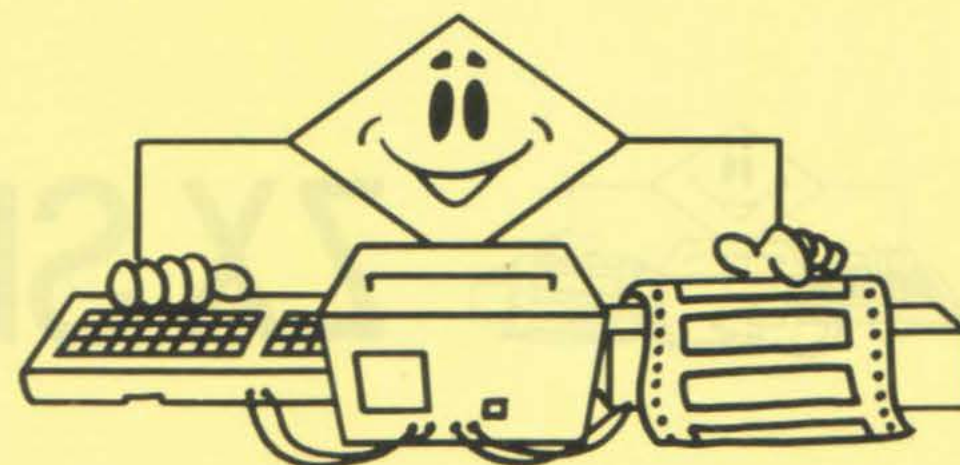
di **G. M. Mellina**

Il primo programma, GRAFCOMP, ha molte possibilità rivolte all'utilizzo dei caratteri grafici ("user defined graphics").
L'uso più semplice è quello che consente di caricare singoli caratteri grafici immettendone gli 8 bit in binario oppure in esadecimale, e verificarne l'immissione mediante la stampa automatica del carattere creato, ingrandito 64 volte.

Una volta creati i caratteri desiderati, è offerta la possibilità di registrarli su cassetta salvandoli come CODE e di verificarne la registrazione.
La possibilità più interessante è però quella che consente di creare "composizioni grafiche" della dimensione desiderata: il limite massimo è dato dall'ampiezza dello schermo (22 righe) per il numero di caratteri grafici consentiti dallo Spectrum (21).

Come conseguenza del punto precedente, il programma offre la possibilità di stampare in qualsiasi momento la composizione parziale o definitiva, centrandola sullo schermo in base alle coordinate che noi abbiamo scelto.

In qualsiasi momento è anche consentito modificare singoli caratteri della composizione e valutarne il risultato tramite stampa della stessa, che in questo caso avviene automaticamente al termine della variazione.
Infine è naturalmente possibile comandare la registrazione su cassetta della videata con la composizione, salvandola come SCREEN$ e/o della schiera, che illustro di seguito, salvandola come DATA, ed eseguirne la verifica.

Per creare una composizione grafica, il programma usa una tabella (schiera), dimensionandola in funzione dei valori da noi immessi in fase di inizializzazione (max 22 righe x 21 colonne): in essa verranno memorizzati, riga per riga, i caratteri grafici che creeremo singolarmente. Quando si richiede la stampa della composizione, ogni singola riga di tabella viene impostata nell'area di memoria "user defined graphics" e stampata, ottenendo così la composizione completa.
Due sole avvertenze: le righe debbono essere create in sequenza dall'alto verso il basso, dalla 1 (riga superiore della composizione e prima riga della tabella) alla n (max 22: ultima riga inferiore della composizione e della tabella); il comando di memorizzazione riga va dato solamente quando tutti i caratteri richiesti in quella riga sono stati creati.

Le singole colonne di ciascuna riga sono identificate con le lettere a ÷ u corrispondenti agli "user defined graphics".

Le istruzioni e l'esemplificazione incorporate nel programma dallo statement 1000 e seguenti penso siano più che sufficienti per chiarire il meccanismo.

Da rilevare inoltre che, una volta inizializzato il programma immettendo i parametri di colonne e righe per il dimensionamento della tabella, è possibile interromperlo con BREAK ed eseguire il LOAD di una schiera precedentemente salvata per potervi apportare modifiche sfruttando tali possibilità di variazione offerte dal programma.

Ciascuno, secondo i propri gusti e le proprie necessità, potrà poi completare a piacere la schermata contenente la composizione, con titoli od altro, ed anche colorarla facilmente eseguendo dei POKE nell'area di memoria corrispondente agli attributi (esempio per paper rosso: FOR n = 22528 TO 23295: POKE n,16: NEXT N).

A titolo di esempio, allego la composizione di una "formula 1" da me creata e magnificamente colorata in rosso, così da ssomigliare ad una rampante Ferrari!

Il programma, caricato su CPU 16 Kbyte, consente la creazione di composizioni limitate (max 32 caratteri: ex. 4 x 8): è però sufficiente cancellare gli statement corrispondenti alle istruzioni (da 1000 a 1130), dopo averli naturalmente trascritti, per sfruttare al massimo le possibilità di composizione.

Il SAVE del programma completo di istruzioni, deve essere eseguito con l'opzione LINE 1000.

Il secondo programma, SLOTMACH, è un divertente giochino d'azzardo, una slot-machine come risulta già evidente dal nome, in cui l'animazione svolge un ruolo predominante.

Per non togliere completamente il piacere della sorpresa, dirò solo che nelle finestrelle della slot-machine possono comparire, a seguito di ogni puntata, in combinazione del tutto casuale, tre delle quattro possibili figure: canarino, mela, limone, dollari.

Naturalmente si può perdere o vincere a seconda della combinazione risultante.

Dirò anche che le mele possono essere bacate, e ciò comporta una penalizzazione... ma il canarino, se può, cercherà di favorirci mangiandosi i dannosi vermetti, dando vita ad una piacevole animazione.

Non resta che tentare la fortuna e sperare nella giusta randomizzazione del vostro Spectrum!

Per gli impazienti o per coloro con l'indice ormai decisamente storto, posso suggerire di apportare temporaneamente le seguenti modifiche al programma:

```
210 LET v=2
1101 LET a=2
1102 LET b=1
1103 LET c=2
```

oppure:

```
LET a=1
LET b=2
LET c=1
LET v=2
```

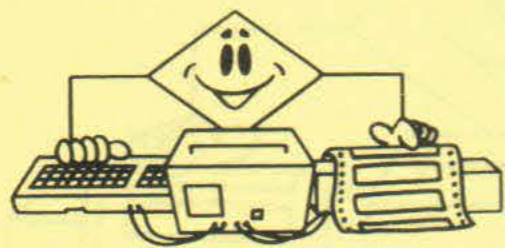
Seguito listato 1.

```
 145 DATA 255,255,255,255,255,25
5,255,255
 146 DATA 255,255,255,255,255,25
5,255,255
 147 DATA 0,0,0,0,0,0,0,0
 148 DATA 0,0,0,0,0,0,0,0
 149 DATA 0,0,0,0,0,0,0,0
 150 DATA 127,15,3,1,0,0,0,0
 151 DATA 0,12,110,251,241,96,0,
0
 152 DATA 0,0,0,0,0,0,3,15
 153 DATA 0,0,2,7,3,1,249,255
 154 DATA 0,0,0,0,0,128,159,255
 155 DATA 0,0,0,0,0,0,192,240
 156 DATA 31,55,109,118,234,213,
234,221
 157 DATA 255,95,183,219,119,191
,255,127
 158 DATA 255,255,255,255,255,25
5,255,255
 159 DATA 248,252,254,254,255,25
5,255,255
 160 DATA 234,221,234,245,234,11
7,118,63
 161 DATA 255,191,255,127,191,95
,191,95
 162 DATA 255,255,255,255,255,25
5,255,255
 163 DATA 255,255,255,255,255,25
4,254,252
 164 DATA 63,31,15,7,1,0,0,0
 165 DATA 171,87,175,255,255,61,
0,0
 166 DATA 255,255,255,255,255,18
8,0,0
 167 DATA 252,248,240,224,128,0,
0,0
 168 DATA 0,0,0,0,195,231,60,24
 169 DATA 0,0,0,0,0,0,0,1
 170 DATA 0,0,0,7,31,123,245,234
 171 DATA 0,0,0,248,252,254,95,1
75
 172 DATA 0,0,0,0,0,16,248,232
 173 DATA 3,7,15,14,29,26,31,58
 174 DATA 213,170,117,170,213,18
6,85,170
 175 DATA 127,191,127,255,127,25
5,255,255
 176 DATA 192,224,240,248,248,24
8,248,248
 177 DATA 61,58,53,58,53,26,21,2
6
 178 DATA 117,205,85,171,87,175,
95,191
 179 DATA 255,255,255,255,255,25
5,255,255
 180 DATA 248,248,248,240,240,22
4,224,192
 181 DATA 30,15,15,15,1,0,0,0
 182 DATA 255,255,255,255,255,31
,0,0
 183 DATA 255,255,254,252,240,12
8,0,0
 184 DATA 128,0,0,0,0,0,0,0
 185 DATA 0,48,118,223,143,6,0,0
```

Volendo aumentare le probabilità di comparsa del canarino e conseguentemente una maggior frequenza delle animazioni, si possono apportare permanentemente le seguenti modifiche:

```
220  LET xx = INT (RND*5): IF xx=4 THEN LET xx=1
221  RETURN
1101 GOSUB 220: LET a=xx: LET po=6:GOSUB 200
1102 GOSUB 220: LET b=xx: LET po=14: GOSUB
     200
1103 GOSUB 220: LET c=xx: LET po=22: GOSUB
     200
```

Per i disegni ho utilizzato il precedente programma GRAFCOMP salvando poi la schiera come CODE: il programma SLOT MACH salvato con LINE 2000, quando viene caricato e va in esecuzione, esegue automaticamente il LOAD della schiera SLOTS allo statement 2010.

Vi fornisco però il programmino che consente di creare la schera mediante DATA: una volta eseguitolo, effettua la registrazione della schiera su cassetta con il comando SAVE "SLOTS" DATA x().

ZX SPECTRUM

Listato 2 - Programma SLOTMACH *versione originale.*

```
  1>REM    ■SLOT-MACHINE■
              © G.M.Mellina
   2 INPUT AT 0,0;"Se vuoi istru
zioni sui punteggi,immetti ""i""
:","==> "; LINE z$: IF z$="i" TH
EN GO TO 1500
  10 BORDER 5: POKE 23693,105: C
LS : POKE 23609,255
  41 FOR n=0 TO 14 STEP 2: PRINT
 AT 0,n;"■■";AT 1,n;"■■";AT 0,n+
16;"■■";AT 1,n+16;"■■";AT 14,n;"
■■";AT 15,n;"■■";AT 14,n+16;"■■"
;AT 15,n+16;"■■": NEXT n
  42 FOR n=1 TO 5 STEP 2: PRINT
AT n,0;"■■";AT n+1,0;"■■";AT n,3
0;"■■";AT n+1,30;"■■";AT n+7,0;"
■■";AT n+8,0;"■■";AT n+7,30;"■■"
;AT n+8,30;"■■": NEXT n
  43 PRINT AT 0,0;"■";AT 0,31;"■
";AT 15,0;"■";AT 15,31;"■";AT 1,
1;"■";AT 1,30;"■";AT 7,0;"■■";AT
 7,30;"■■";AT 14,0;"■■";AT 14,30
;"■■"
  50 FOR z=0 TO 4: FOR n=0 TO 7:
 POKE USR "q"+z*8+n,x(z+1,n+129)
: NEXT n: NEXT z
  60 GO SUB 106: GO SUB 102: GO
TO 1000
 100 POKE 23693,66: LET z$="$$$$
": GO TO 103
 101 POKE 23693,87: LET z$="????
": GO TO 103
 102 LET po=6: GO SUB 100: LET p
o=14: GO SUB 100: LET po=22: GO
SUB 100: RETURN
 103 FOR n=6 TO 9: PRINT AT n,po
;z$: BEEP .1,15-n: NEXT n: RETUR
N
 105 POKE 23693,104: PRINT AT 7,
10;"    ";AT 7,18;"    ": RETURN

 106 POKE 23693,40: PRINT AT 16,
0;" ";: FOR n=1 TO 191: PRINT "
";: NEXT n: RETURN
 110 POKE 23693,col: FOR m=0 TO
3: LET z=144+m*4: PRINT AT 6+m,p
o;CHR$ (z);CHR$ (z+1);CHR$ (z+2)
;CHR$ (z+3): BEEP .1,m*2+12: NEX
T m: RETURN
 111 LET z=1: LET col=70: LET pk
=1: GO TO 120
 112 LET z=4: LET col=114: LET p
k=2: GO TO 120
 113 LET z=5: LET col=102: LET p
k=3: GO TO 120
 114 LET z=2: LET col=70: GO TO
120
 115 LET z=3: LET col=70
 120 FOR n=0 TO 127: POKE USR "a
"+n,x(z,n+1): NEXT n: RETURN
 121 POKE 23693,104: PRINT AT 7,
vo+1;"■": GO SUB 125: PRINT AT 7
,vo;"■■": GO TO 125
 122 POKE 23693,104: PRINT AT 7,
vo;"■": GO SUB 125: PRINT AT 7,v
o;"■■": GO TO 125
 123 POKE 23693,104: PRINT AT 7,
vo;"■■";AT 6,vo;"  ";AT 8,vo;"
";AT 9,vo;"  ": RETURN
 124 POKE 23693,104: PRINT AT 7,
vo;"■■";AT 6,vo;"  ";AT 8,vo;"
";AT 9,vo;"  ": RETURN
 125 BEEP .1,23: PAUSE 15: RETUR
N
 131 PRINT PAPER 0; INK 6; FLASH
 1;AT 7,po+2;"■": GO SUB 114: GO
 SUB 110: PAUSE 10
 132 FOR m=0 TO 1: LET y=po+m: L
ET z=1+m*4: FOR n=0 TO 3: PRINT
AT 6+n,y;c$(z+n): NEXT n: NEXT m
: FOR m=0 TO 3: LET y=po+m+2: LE
T z=144+m: FOR n=0 TO 3: PRINT A
T 6+n,y;CHR$ (z+n*4): NEXT n: NE
XT m
 133 PAUSE 10: LET vo=po+4: PRIN
T PAPER 5;AT 7,vo+2;"  ": GO SUB
 123: LET col=70: GO SUB 110
 134 GO SUB 111: PRINT PAPER 5;
INK 0;AT 7,vo;"■ ": GO SUB 136:
PAUSE 10: PRINT PAPER 5;AT 7,vo;
"  ": GO SUB 136
 135 PAUSE 10: GO SUB 110: RETUR
N
 136 PRINT FLASH 1;AT 9,po;"■":
FOR n=1 TO 3: BEEP .1,12: BEEP .
05,-12: BEEP .05,0: NEXT n: RETU
RN
 141 PRINT PAPER 0; INK 6; FLASH
 1;AT 7,po+1;"■": GO SUB 115: GO
 SUB 110: PAUSE 10
 142 FOR m=0 TO 1: LET y=po+m+2:
 LET z=1+m*4: FOR n=0 TO 3: PRIN
T AT 6+n,y;c$(z+n): NEXT n: NEXT
 m: FOR m=3 TO 0 STEP -1: LET y=
```

Le pause che si avvertono durante l'esecuzione del gioco e che purtroppo ne rallentano un poco il dinamismo, sono dovute al tempo, inspiegabilmente lungo, che lo Spectrum impiega per eseguire il POKE dei 16 carateri grafici necessari per visualizzare ogni figura. Occorrerebbe scrivere la routinetta (statement 120) direttamente in linguaggio macchina e il problemino sarebbe risolto.

La versione originale del programma non può girare su CPU 16 Kbyte, ma anche qui piccole modifiche che non intaccano minimamente l'essenza e l'animazione del gioco risolvono il problema.

Occorre eliminare le istruzioni (statement 1500 ÷ 1530) e gli statement che stampano la cornice del riquadro (41 ÷ 43) e "pokano" i caratteri grafici q-r-s-t-u (50) e modificare i seguenti statement:

```
2000 REM DIM x(5,128)
2010 CLS: PRINT AT 3,2; FLASH 1; "LOADING DATA
     e CODE": LOAD "SLOTS" DATA x(): LOAD
     "SLOTG" CODE USR "q" GO TO 1
```

Naturalmente occorre registrare su cassetta i due blocchi di dati: vi fornisco perciò il programmino che consente di creare la schiera ed il POKE dei 5 caratteri grafici mediante DATA: una volta eseguitolo, effettua la registrazione di schiera e grafici con i comandi SAVE "SLOTS" DATA x() e SAVE "SLOTG" CODE USR "q", 40.

In tutti i casi il save del programma SLOTMACH deve essere eseguito col l'opzione LINE 2000. ■

Seguito listato 2.

```
po+m-2: LET z=144+m: FOR n=0 TO
3: PRINT AT 6+n,y;CHR$ (z+n*4):
NEXT n: NEXT m
 143 PAUSE 10: LET vo=po-2: PRIN
T PAPER 5;AT 7,vo-2;"  ": GO SUB
 124: LET col=70: GO SUB 110
 144 GO SUB 111: PRINT PAPER 5;
INK 0;AT 7,vo;" ■": GO SUB 146:
PAUSE 10: PRINT PAPER 5;AT 7,vo+
1;"  ": GO SUB 146
 145 PAUSE 10: GO SUB 110: RETUR
N
 146 PRINT FLASH 1;AT 9,po+3;"■"
: FOR n=1 TO 3: BEEP .1,12: BEEP
 .05,-12: BEEP .05,0: NEXT n: RE
TURN
 200 GO SUB 101: IF xx=0 THEN GO
 SUB 100: GO TO 204
 201 IF pk=xx THEN GO TO 203
 202 GO SUB k(xx)
 203 GO SUB 110
 204 RETURN
 210 LET v=INT (RND*5)
 211 IF xx=1 AND v>1 THEN LET va
=1: LET vo=10: GO SUB 122: GO TO
 216
 212 IF xx=3 AND v<3 THEN LET vc
=1: LET vo=20: GO SUB 121: GO TO
 216
 213 IF xx<>2 THEN GO TO 216
 214 IF v<3 THEN LET vb=1: LET v
o=12: GO SUB 121
 215 IF v>1 THEN LET vb=vb+1: LE
T vo=18: GO SUB 122
 216 RETURN
1000 POKE 23693,40: PRINT AT 17,
0;"La  Direzione  del  Casino'
ti elargisce  un  bonus  di  $ 1
00 perche' tu tenti la Dea benda
ta!": INPUT AT 0,0;"Fai la tua p
untata:  ==> ";pnt
1010 GO SUB 106: LET cap=100
1100 LET pk=0: LET ris=0
1101 LET a=INT (RND*4): LET po=6
: LET xx=a: GO SUB 200
1102 LET b=INT (RND*4): LET po=1
4: LET xx=b: GO SUB 200
1103 LET c=INT (RND*4): LET po=2
```

Seguito listato 2.

```
0: LET xx=c: GO SUB 200
1110 IF a=b AND a=c THEN GO TO 1
150
1111 IF a=0 AND (b=0 OR c=0) THE
N GO TO 1151
1112 IF b=0 AND c=0 THEN GO TO 1
151
1113 IF a=0 AND b=c THEN GO TO 1
151
1114 IF b=0 AND a=c THEN GO TO 1
151
1115 IF c=0 AND a=b THEN GO TO 1
151
1121 LET v=0. LET va=0. IF a=2 T
HEN LET xx=1: GO SUB 210
1122 LET vc=0: IF c=2 THEN LET x
x=3: GO SUB 210
1123 LET vb=0: IF b=2 THEN LET x
x=2: GO SUB 210
1130 LET vm=0
1131 IF va=1 AND b=1 THEN LET po
=14: GO SUB 141: LET vm=1
1132 IF vc=1 AND b=1 THEN LET po
=14: GO SUB 131: LET vm=vm+1  GO
 TO 1140
1133 IF a=1 AND vb>0 AND v<3 THE
N LET po=6: GO SUB 131: LET vm=1
1134 IF c=1 AND vb>0 AND v>1 THE
N LET po=22: GO SUB 141: LET vm=
vm+1
1140 IF vm=2 THEN LET vm=3
1141 LET ris=-pnt-(va+vb+vc)*pnt
/2+vm*pnt
1142 GO TO 1160
1150 IF a=0 THEN LET ris=pnt
1151 LET ris=ris+pnt*2
1160 LET cap=cap+ris
1161 PAPER 5: INK 0: LET r$="vin
to": IF ris<0 THEN LET r$="perso
": LET ris=ris*-1
1162 POKE 23693,40: PRINT AT 17,
0;"Hai puntato","$ ";pnt," e ";
r$,"$ ";ris
1163 IF cap<=0 THEN PRINT "hai r
ischiato troppo","e sei finito s
ul sul lastrico !!!": LET pnt=0:
 GO TO 1171
1164 PRINT "il tuo capitale ammo
nta ora a","$ ";cap," "
1170 INPUT AT 0,0;"Fai un'altra
puntata (""0"" se lasci): ==> "
;pnt: GO SUB 106
1171 GO SUB 105: IF pnt=0 THEN G
O SUB 102: POKE 23609,0: BORDER
7: POKE 23693,56: STOP : GO TO 1
1172 GO TO 1100
1500 POKE 23693,56: CLS : PRINT
CHR$ 13;"▌PUNTEGGIO FIGURE:","
- 3 figure uguali        = +2","-
 3 figure ""$""          = +3","
- 2 figure ""$""         = +2","
 - 2 fig.ug.+ 1 fig.""$"" = +2",
" - 3 figure diverse     = -1"
1510 PRINT CHR$ 13;CHR$ 13;"▌PU
NTEGGIO VERMI:"," - mele bacate
  = -1/2 per","     ogni verme","
 - vermi mangiati =","    +1 X un
 verme",,"o +3 X due vermi"
1520 PRINT CHR$ 13;CHR$ 13;"▌PU
NTEGGIO TOTALE:","     = puntata
 X punti(+/-)"
1530 PRINT BRIGHT 1;AT 21,0;"Pre
mi ENTER per giocare": PAUSE 0:
GO TO 10
2000 REM DIM x(5,136)
2005 LET c$="▀▀ ▀▀". DIM k(3)
: LET k(1)=111: LET k(2)=112: LE
T k(3)=113
2010 CLS : PRINT FLASH 1;AT 3,2;
"LOAD ""SLOTS"" IN ESECUZIONE":
LOAD "SLOTS" DATA x(): GO TO 1
1424 REM © G.M.Mellina ©
```

ZX SPECTRUM

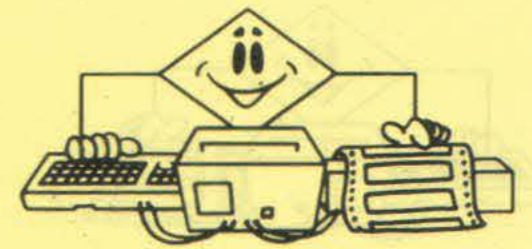
Listato 3 - Programma per caricare e registrare schiera ridotta e grafici per SLOTMATCH ridotto per 16 Kbyte.

```
  10 DIM x(5,128)
  20 FOR z=1 TO 5: FOR m=1 TO 12
8
  30 READ x(z,m)
  40 NEXT m: NEXT z
  50 FOR n=USR "q" TO USR "u"+7:
 READ g: POKE n,g: NEXT n
  60 SAVE "SLOTS" DATA x()
  70 SAVE "SLOTG" CODE USR "q":
STOP
 101 DATA 0,0,0,0,0,0,1,3
 102 DATA 0,0,3,31,127,255,255,2
55
 103 DATA 0,0,192,248,254,255,25
5,255
 104 DATA 0,0,0,0,0,0,128,192
 105 DATA 7,7,15,15,31,31,63,63
 106 DATA 255,143,7,7,7,143,255,
254
 107 DATA 255,241,224,224,224,24
1,255,127
 108 DATA 224,224,240,240,248,24
8,252,252
 109 DATA 63,63,63,63,63,63,63,6
3
 110 DATA 252,249,243,231,207,15
9,175,3
 111 DATA 63,159,207,231,243,249
,245,192
 112 DATA 252,252,252,252,252,25
2,252,252
 113 DATA 63,31,31,15,7,3,1,0
 114 DATA 241,254,255,255,255,25
5,255,63
 115 DATA 143,127,255,255,255,25
5,255,252
 116 DATA 252,248,248,240,224,19
2,128,0
 117 DATA 0,0,0,0,3,7,15,31
 118 DATA 0,0,63,255,255,255,255
,254
 119 DATA 0,0,128,240,252,254,25
5,63
 120 DATA 0,0,0,0,0,0,0,0
 121 DATA 31,63,127,127,255,255,
255,255
 122 DATA 252,252,252,254,255,25
5,255,255
 123 DATA 31,31,31,63,248,255,25
5,255
 124 DATA 192,248,255,0,0,255,24
8,192
 125 DATA 255,255,255,255,255,25
5,255,255
 126 DATA 255,255,255,255,255,25
5,255,255
 127 DATA 255,255,254,252,248,24
0,224,0
 128 DATA 0,0,0,0,0,0,0,0
 129 DATA 254,240,192,128,0,0,0,
0
 130 DATA 0,0,0,0,0,0,0,0
 131 DATA 0,0,0,0,0,0,0,0
 132 DATA 0,0,0,0,0,0,0,0
 133 DATA 0,0,0,0,0,0,0,0
 134 DATA 0,0,1,15,63,127,255,25
2
 135 DATA 0,0,252,255,255,255,25
5,127
 136 DATA 0,0,0,0,192,224,240,24
8
 137 DATA 3,31,255,0,0,255,31,3
 138 DATA 248,248,248,252,31,255
,255,255
 139 DATA 63,63,63,127,255,255,2
55,255
 140 DATA 248,252,254,254,255,25
5,255,255
 141 DATA 0,0,0,0,0,0,0,0
 142 DATA 255,255,127,63,31,15,7
,0
 143 DATA 255,255,255,255,255,25
5,255,255
 144 DATA 255,255,255,255,255,25
5,255,255
 145 DATA 0,0,0,0,0,0,0,0
 146 DATA 0,0,0,0,0,0,0,0
 147 DATA 0,0,0,0,0,0,0,0
 148 DATA 127,15,3,1,0,0,0,0
 149 DATA 0,0,0,0,0,0,3,15
 150 DATA 0,0,2,7,3,1,249,255
 151 DATA 0,0,0,0,0,128,159,255
 152 DATA 0,0,0,0,0,0,192,240
 153 DATA 31,55,109,118,234,213,
234,221
 154 DATA 255,95,183,219,119,191
,255,127
 155 DATA 255,255,255,255,255,25
5,255,255
 156 DATA 248,252,254,254,255,25
5,255,255
 157 DATA 234,221,234,245,234,11
```

Seguito listato 3.

```
7,118,63
 158 DATA 255,191,255,127,191,95
,191,95
 159 DATA 255,255,255,255,255,25
5,255,255
 160 DATA 255,255,255,255,255,25
4,254,252
 161 DATA 63,31,15,7,1,0,0,0
 162 DATA 171,87,175,255,255,61,
0,0
 163 DATA 255,255,255,255,255,18
8,0,0
 164 DATA 252,248,240,224,128,0,
0,0
 165 DATA 0,0,0,0,0,0,0,1
 166 DATA 0,0,0,7,31,123,245,234
 167 DATA 0,0,0,248,252,254,95,1
75
 168 DATA 0,0,0,0,0,16,248,232
 169 DATA 3,7,15,14,29,26,31,58
 170 DATA 213,170,117,170,213,18
6,85,170
 171 DATA 127,191,127,255,127,25
5,255,255
 172 DATA 192,224,240,248,248,24
8,248,248
 173 DATA 61,58,53,58,53,26,21,2
6
 174 DATA 117,205,85,171,87,175,
95,191
 175 DATA 255,255,255,255,255,25
5,255,255
 176 DATA 248,248,248,240,240,22
4,224,192
 177 DATA 30,15,15,15,1,0,0,0
 178 DATA 255,255,255,255,255,31
,0,0
 179 DATA 255,255,254,252,240,12
8,0,0
 180 DATA 128,0,0,0,0,0,0,0
 201 DATA 255,143,31,31,31,143,2
55,255
 202 DATA 255,241,248,248,248,24
1,255,255
 203 DATA 0,12,110,251,241,96,0,
0
 204 DATA 0,0,0,0,195,231,60,24
 205 DATA 0,48,118,223,143,6,0,0
```

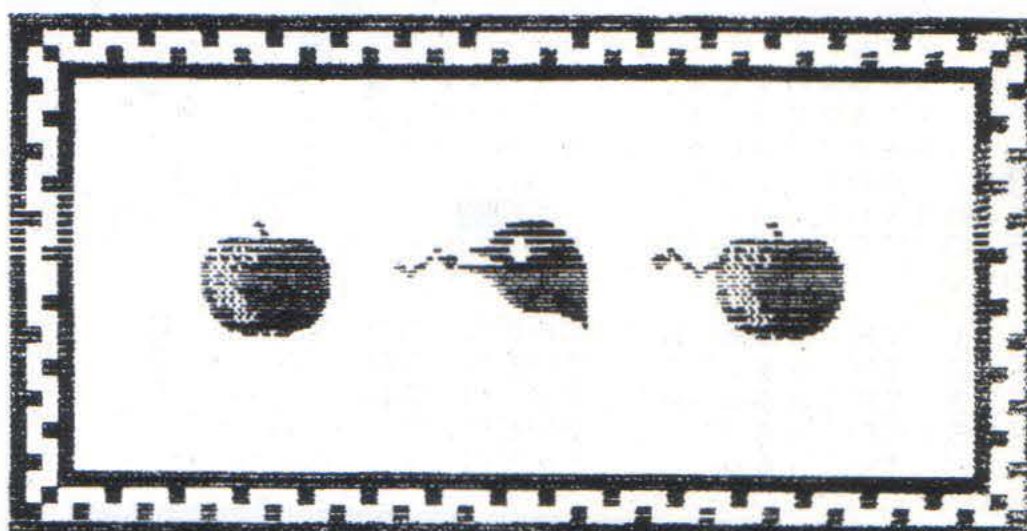
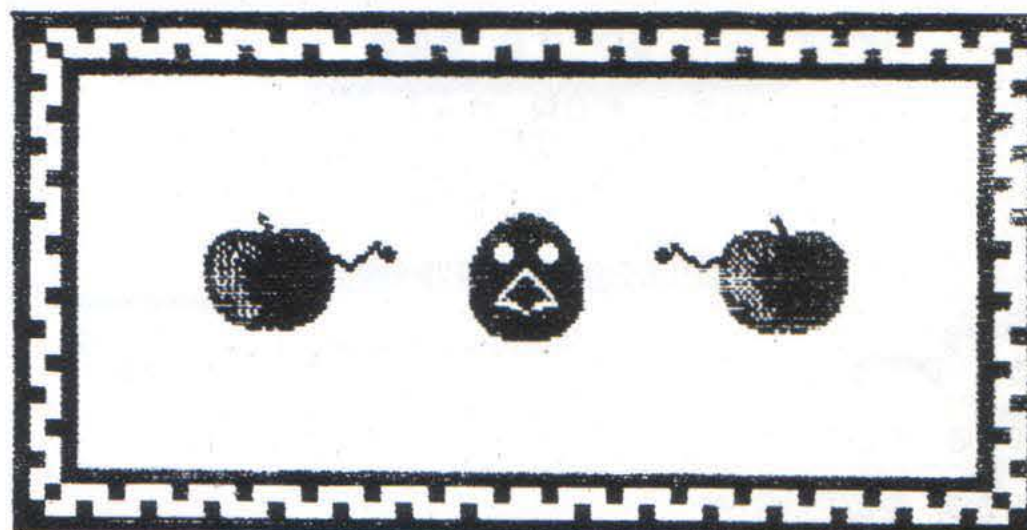
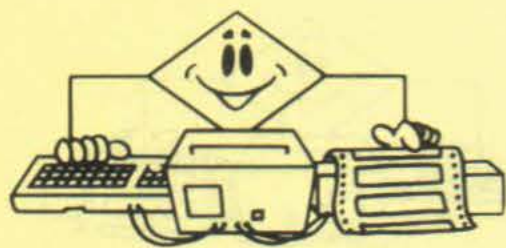
Listato 4 - Programma SLOTMACR, versione 16 Kbyte di SLOTMACH.

```
  10 BORDER 5: POKE 23693,105: C
LS : POKE 23609,255
  60 GO SUB 106: GO SUB 102: GO
TO 1000
 100 POKE 23693,66: LET z$="$$$$
": GO TO 103
 101 POKE 23693,87: LET z$="????
": GO TO 103
 102 LET po=6: GO SUB 100: LET p
o=14: GO SUB 100: LET po=22: GO
SUB 100: RETURN
 103 FOR n=6 TO 9: PRINT AT n,po
;z$: BEEP .1,15-n: NEXT n: RETUR
N
 105 POKE 23693,104: PRINT AT 7,
10;"    ";AT 7,18;"    ": RETURN

 106 POKE 23693,40: PRINT AT 15,
0;" ";: FOR n=1 TO 191: PRINT "
";: NEXT n: RETURN
 110 POKE 23693,col: FOR m=0 TO
3: LET z=144+m*4: PRINT AT 6+m,p
o;CHR$ (z);CHR$ (z+1);CHR$ (z+2)
;CHR$ (z+3): BEEP .1,m*2+12: NEX
T m: RETURN
 111 LET z=1: LET col=70: LET pk
=1: GO TO 120
 112 LET z=4: LET col=114: LET p
k=2: GO TO 120
 113 LET z=5: LET col=102: LET p
k=3: GO TO 120
 114 LET z=2: LET col=70: GO TO
120
 115 LET z=3: LET col=70
 120 FOR n=0 TO 127: POKE USR "a
"+n,x(z,n+1): NEXT n: RETURN
 121 POKE 23693,104: PRINT AT 7,
vo;" [UDG]": GO SUB 125: PRINT AT 7,
vo;"[UDG]": GO TO 125
 122 POKE 23693,104: PRINT AT 7,
vo;"[UDG]": GO SUB 125: PRINT AT 7,v
o;"[UDG]": GO TO 125
 123 POKE 23693,104: PRINT AT 7,
vo;"[UDG]";AT 6,vo;"  ";AT 8,vo;"
```

Seguito listato 4.

```
";AT 9,vo;"  ": RETURN
 124 POKE 23693,104: PRINT AT 7,
vo;"■";AT 6,vo;"  ";AT 8,vo;"
";AT 9,vo;"  ": RETURN
 125 BEEP .1,23: PAUSE 15: RETUR
N
 131 PRINT PAPER 0; INK 6; FLASH
 1;AT 7,po+2;"■": GO SUB 114: GO
 SUB 110: PAUSE 10
 132 FOR m=0 TO 1: LET y=po+m: L
ET z=1+m*4. FOR n=0 TU 3: PRINT
AT 6+n,y;c$(z+n): NEXT n: NEXT m
: FOR m=0 TO 3: LET y=po+m+2: LE
T z=144+m: FOR n=0 TO 3: PRINT A
T 6+n,y;CHR$ (z+n*4): NEXT n: NE
XT m
 133 PAUSE 10: LET vo=po+4: PRIN
T PAPER 5;AT 7,vo+2;"  ": GO SUB
 123: LET col=70: GO SUB 110
 134 GO SUB 111: PRINT PAPER 5;
INK 0;AT 7,vo;"■ ": GO SUB 136:
PAUSE 10: PRINT PAPER 5;AT 7,vo;
"  ": GO SUB 136
 135 PAUSE 10: GO SUB 110: RETUR
N
 136 PRINT FLASH 1;AT 9,po;"■":
FOR n=1 TO 3: BEEP .1,12: BEEP .
05,-12: BEEP .05,0: NEXT n: RETU
RN
 141 PRINT PAPER 0; INK 6; FLASH
 1;AT 7,po+1;"■": GO SUB 115: GO
 SUB 110: PAUSE 10
 142 FOR m=0 TO 1: LET y=po+m+2:
 LET z=1+m*4: FOR n=0 TO 3: PRIN
T AT 6+n,y;c$(z+n): NEXT n: NEXT
 m: FOR m=3 TO 0 STEP -1: LET y=
po+m-2: LET z=144+m: FOR n=0 TO
3: PRINT AT 6+n,y;CHR$ (z+n*4):
NEXT n: NEXT m
 143 PAUSE 10: LET vo=po-2: PRIN
T PAPER 5;AT 7,vo-2;"  ": GO SUB
 124: LET col=70: GO SUB 110
 144 GO SUB 111: PRINT PAPER 5,
INK 0;AT 7,vo;" ■": GO SUB 146:
PAUSE 10: PRINT PAPER 5;AT 7,vo+
1;"  ": GO SUB 146
 145 PAUSE 10: GO SUB 110: RETUR
N
 146 PRINT FLASH 1;AT 9,po+3;"■"
: FOR n=1 TO 3: BEEP .1,12: BEEP
 .05,-12: BEEP .05,0: NEXT n: RE
 200 GO SUB 101: IF xx=0 THEN GO
 SUB 100: GO TO 204
 201 IF pk=xx THEN GO TO 203
 202 GO SUB k(xx)
 203 GO SUB 110
 204 RETURN
 210 LET v=INT (RND*5)
 211 IF xx=1 AND v>1 THEN LET va
=1: LET vo=10: GO SUB 122: GO TO
 216
 212 IF xx=3 AND v<3 THEN LET vc
=1: LET vo=20: GO SUB 121: GO TO
 216
 213 IF xx<>2 THEN GO TO 216
 214 IF v<3 THEN LET vb=1: LET v
o=12: GO SUB 121
 215 IF v>1 THEN LET vb=vb+1. LE
T vo=18: GO SUB 122
 216 RETURN
1000 POKE 23693,40: PRINT AT 17,
0;"La  Direzione  del  Casino'
ti elargisce  un  bonus  di  $ 1
00 perche' tu tenti la Dea benda
ta!": INPUT AT 0,0;"Fai la tua p
untata:  ==> ";pnt
1010 GO SUB 106: LET cap=100
1100 LET pk=0: LET ris=0
1101 LET a=INT (RND*4): LET po=6
: LET xx=a: GO SUB 200
1102 LET b=INT (RND*4): LET po=1
4: LET xx=b: GO SUB 200
1103 LET c=INT (RND*4): LET po=2
2: LET xx=c: GO SUB 200
1110 IF a=b AND a=c THEN GO TO 1
150
1111 IF a=0 AND (b=0 OR c=0) THE
N GO TO 1151
1112 IF b=0 AND c=0 THEN GO TO 1
151
1113 IF a=0 AND b=c THEN GO TO 1
151
1114 IF b=0 AND a=c THEN GO TO 1
151
1115 IF c=0 AND a=b THEN GO TO 1
151
1121 LET v=0: LET va=0: IF a=2 T
HEN LET xx=1: GO SUB 210
1122 LET vc=0: IF c=2 THEN LET x
x=3: GO SUB 210
```

Seguito listato 4.

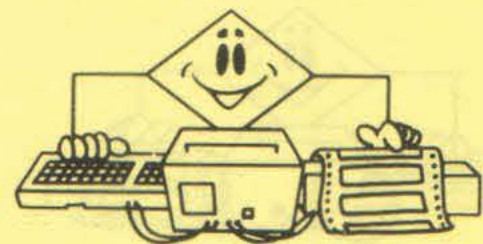
```
1123 LET vb=0: IF b=2 THEN LET x
x=2: GO SUB 210
1130 LET vm=0
1131 IF va=1 AND b=1 THEN LET po
=14: GO SUB 141: LET vm=1
1132 IF vc=1 AND b=1 THEN LET po
=14: GO SUB 131: LET vm=vm+1: GO
 TO 1140
1133 IF a=1 AND vb>0 AND v<3 THE
N LET po=6: GO SUB 131: LET vm=1
1134 IF c=1 AND vb>0 AND v>1 THE
N LET po=22: GO SUB 141: LET vm=
vm+1
1140 IF vm=2 THEN LET vm=3
1141 LET ris=-pnt-(va+vb+vc)*pnt
/2+vm*pnt
1142 GO TO 1160
1150 IF a=0 THEN LET ris=pnt
1151 LET ris=ris+pnt*2
1160 LET cap=cap+ris
1161 PAPER 5: INK 0: LET r$="vin
to": IF ris<0 THEN LET r$="perso
": LET ris=ris*-1
1162 POKE 23693,40: PRINT AT 17,
0;"Hai puntato","$ ";pnt,"  e ";
r$,"$ ";ris
1163 IF cap<=0 THEN PRINT "hai r
ischiato troppo","e sei finito s
ul lastrico !!!": LET pnt=0: GO
TO 1171
1164 PRINT "il tuo capitale ammo
nta ora a","$ ";cap,"  "
1170 INPUT AT 0,0;"Fai un'altra
puntata (""0"" se lasci):  ==> "
;pnt: GO SUB 106
1171 GO SUB 105: IF pnt=0 THEN G
O SUB 102: POKE 23609,0: BORDER
7: POKE 23693,56: STOP : GO TO 1
1172 GO TO 1100
2000 REM DIM x(5,128)
2005 LET c$=" ■■ ■■": DIM k(3)
: LET k(1)=111: LET k(2)=112: LE
T k(3)=113
2010 CLS : PRINT AT 3,2; FLASH 1
;"LOADING DATA e CODE ": LOAD "S
LOTS" DATA x(): LOAD "SLOTS"CODE
 : GO TO 1
```

Listato 5 - COPY *di due screen di* SLOTMACH.

ZX SPECTRUM

Listato 6 - Programma GRAFCOMP.

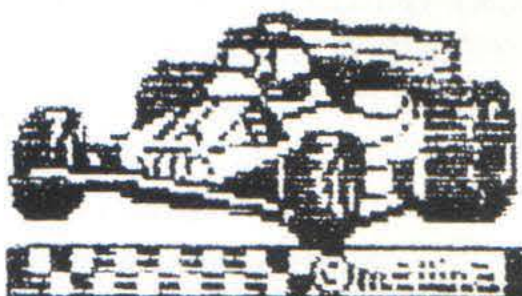
```
   1>REM  COMPOSIZIONI GRAFICHE 
              © G.M.MELLINA
   2 LET var=0: INPUT "Se vuoi c
omporre uno ""screen"" rispond
i "" "" ,altrimenti ""ENTER"":";C
HR$ 13;" ==> "; LINE c$
   3 IF c$<>"s" THEN GO TO 10
   4 INPUT "Immetti il n.di righ
e (rr), e dicolonne (cc) della c
omposizione,e le coordinate (AT
xx,yy) dell'angolo super.sinistr
o di stampa:";CHR$ 13;" rr ==> "
;rr,"cc ==> ";cc," xx ==> ";xx,"
yy ==> ";yy
   5 DIM a(rr,cc*8): LET aa=1
   6 GO SUB 650
  10 INPUT "Vuoi usare il modo b
inario ("" "")o esadecimale ("" 
"") ?";CHR$ 13;" b/e ==> "; LINE
 m$
  20 IF m$="e" THEN GO TO 100
  30 IF m$="b" THEN GO TO 200
  40 GO TO 10
 100 REM ESADECIMALE
 110 CLS : INPUT "Immetti il sim
bolo grafico e gli8 bytes espres
si in esadecimale: (es.: g 3c 42
81 80 8f 81 41 3e)";CHR$ 13;" 
 "; LINE s$
 120 GO SUB 400
 130 FOR s=3 TO 24 STEP 3
 140 LET x=CODE s$(s)-48: IF s$(
s)>"9" THEN LET x=x-39
 150 LET y=CODE s$(s+1)-48: IF s
$(s+1)>"9" THEN LET y=y-39
 160 POKE USR s$(1)+s/3-1,x*16+y
 170 PRINT INK 2;AT 2+s/3,5;" ";
: GO SUB 300: LET x=y: GO SUB 30
0
 180 NEXT s
 190 GO TO 500
 200 REM BINARIO
 210 CLS : INPUT "Immetti il sim
bolo grafico e gli8 bytes espres
si in binario:";CHR$ 13;"(es.:
 g 00111100 00100010 ...)";CHR$
13;"  ==> "; LINE s$
 220 GO SUB 400: FOR s=3 TO 66 S
TEP 9: LET y=0
 230 FOR k=0 TO 7 STEP 4: LET z=
8: LET x=y: LET y=0
 240 FOR b=0 TO 3
 250 IF s$(s+k+b)="1" THEN LET y
=y+z
 260 LET z=z/2: NEXT b: NEXT k
 270 POKE USR s$(1)+(s+6)/9-1,x*
16+y
 280 PRINT INK 2;AT 2+(s+6)/9,5;
" ";: GO SUB 300: LET x=y: GO SU
B 300
 290 NEXT s: GO TO 500
 300 REM ST. CARATT.INGRANDITO
 310 LET c=8: FOR n=1 TO 4
 320 LET x=x-c: IF x>=0 THEN PRI
NT INVERSE 1;"O";: GO TO 340
 330 LET x=x+c: PRINT " ";
 340 LET c=c/2: NEXT n: RETURN
 400 REM STAMPA CORNICI
 410 INK 2: PRINT AT 2,5;"
 ";AT 11,5;"          ";AT 5,
20;"   ";AT 6,20;"   ";AT 7,20;"
   "
 420 FOR n=3 TO 10: PRINT AT n,5
;" ";AT n,14;" ": NEXT n: INK 0
 430 RETURN
 500 REM STAMPA CAR.GRAFICO
 510 PRINT AT 0,5;"carattere gra
f.: ";"""";s$(1);"""";AT 6,21;CH
R$ (CODE s$+47)
 520 IF var=1 THEN GO TO 950
 525 INPUT AT 0,0;"Immetti "" 
"" per terminare,",""" m"" per me
morizzare la riga,",""" s"" per S
AVE graphics o schiera,","""p""
per stampare la comp.graf.",""""
c"" per cambiare modo (b/e)",""
"i"" per istruzioni,","ENTER per
 continuare:"," ==> "; LINE a$
 530 IF a$="c" THEN CLS : GO TO
9999
 531 IF a$="m" THEN GO TO 600
 532 IF a$="s" THEN GO TO 560
 533 IF a$="p" THEN GO TO 700
 535 IF a$="i" THEN GO TO 1010
 536 IF a$="c" THEN GO TO 10
 540 GO TO 20
 560 REM SAVE US.GRAPH./SCHIERA
 565 INPUT "Immetti "" "" per SA
VE graphics,          o "" "" per
SAVE schiera:"," ==> "; LINE t$
```

Seguito listato 6.

```
 570 INPUT "Immetti il nome per
""SAVE"":"," ==> ";a$
 575 IF t$="s" THEN GO TO 590
 580 CLS : PRINT AT 2,0; FLASH 1
;"SAVE """;a$;""" CODE USR ""a""
";AT 4,5;"in esecuzione": FLASH
0: SAVE a$ CODE USR "a",168: CLS
 : PRINT AT 2,0;" SAVE  OK"
 585 FLASH 1: PRINT AT 5,0;"REWI
ND THE TAPE FOR VERIFY": FLASH 0
: VERIFY a$ CODE : CLS : PRINT A
T 2,0;" VERIFY  OK": GO TO 520
 590 CLS : PRINT AT 2,0; FLASH 1
;"SAVE """;a$;""" DATA a()";AT 4
,3;"in esecuzione": FLASH 0: SAV
E a$ DATA a(): CLS : PRINT AT 2,
0;" SAVE  OK"
 595 FLASH 1: PRINT AT 5,0;"REWI
ND THE TAPE FOR VERIFY": FLASH 0
: VERIFY a$ DATA a(): CLS : PRIN
T AT 2,0;" VERIFY  OK": GO TO 5
20
 600 REM MEMORIZZA CAR.GRAFICI
 610 FOR d=1 TO cc*8: LET a(aa,d
)=PEEK (USR "a"+d-1): NEXT d
 620 LET aa=aa+1: GO SUB 650: CL
S : GO TO 520
 650 REM AZZERAMENTO CAR.GRAFICI
 660 FOR d=0 TO cc*8-1: POKE USR
"a"+d,BIN 0: NEXT d: RETURN
 700 REM STAMPA COMPOSIZ.GRAFICA
 710 CLS : FOR n=0 TO rr-1
 720 FOR d=0 TO cc*8-1: POKE USR
"a"+d,a(n+1,d+1): NEXT d
 730 LET r$="ABCDEFGHIJKLMNOPQRS
TU"
 740 PRINT AT xx+n,yy;r$( TO cc)
: NEXT n
 750 INPUT AT 0,0;"Imm. "" "" p
er SAVE SCR.,"," "" "" per vari.1 car
.","ENTER per cont.: "; LINE a$
 760 IF a$="s" THEN GO TO 800
 765 IF a$="v" THEN GO TO 900
 770 GO SUB 650: CLS : GO TO 520
 800 REM SAVE SCREEN
 810 INPUT AT 0,0;"Immetti nome
per ""SAVE SCREEN$"":";CHR$ 13;"
==> ";a$
 820 SAVE a$ SCREEN$ : SAVE a$ S
CREEN$
 830 INPUT "Riavvolgi il nastro
per VERIFY  (LOAD) e premi ENTER
"; LINE a$: LOAD a$ SCREEN$
 840 INPUT "Se TAPE LOAD.ERROR p
uoi ripeterestampa e save.   Pre
mi ENTER "; LINE a$
 850 GO TO 520
 900 REM VARIA UN CAR.DEL QUADRO
 910 LET aas=aa: LET var=1
 920 INPUT AT 0,0;"Immetti il nu
mero della riga delquadro che vu
oi modificare: ";aa
 930 GO TO 20
 950 PAUSE 50: LET car=(CODE s$(
1)-97)*8
 960 FOR d=1 TO 8: LET a(aa,car+
d)=PEEK (USR "a"+car-1+d): NEXT
d
 970 LET aa=aas: LET var=0: GO T
O 700
1000 REM ISTRUZIONI
1001 INPUT "Immetti ""i"" se vuo
i istruzioni, altrimenti ENTER
:";CHR$ 13;" ==> "; LINE a$
1010 CLS : IF a$<>"i" THEN GO TO
 1
1020 PRINT "Il programma consent
e di:";CHR$ 13;" - immettere car
atteri grafici  in binario o esa
decimale, e memorizzarli abbinat
i ad una lettera(a-";CHR$ 8; OVE
R 1;":u)"
1030 PRINT " - comporre ""screen
s"" grafici diformato a piacere
(max 22 righe x 21 colonne), imm
ettendo i sin-goli caratteri gra
fici che com- pongono ogni riga
e stampandoli infine come unica
videata"
1040 PRINT " - memorizzare ogni
singola rigain apposita tabella"
;CHR$ 13;" - stampare la composi
zione gra-fica";CHR$ 13;" - modi
ficare singoli caratteri della c
omposizione";CHR$ 13;" - eseguir
e SAVE SCREEN$ e VERI-FY della c
omposizione  (il SAVE viene effe
ttuato 2 volte)"
1050 PRINT " - eseguire SAVE e V
ERIFY della area di memoria ""us
```

Seguito listato 6.

```
er graphics""";CHR$ 13;" - esegu
ire SAVE e VERIFY della tabella
(schiera ""a"")"
1060 PRINT CHR$ 13;CHR$ 13;"Per
ottenere una composizione graf
ica e' richiesta l'immissio-ne d
i:";CHR$ 13;" - numero delle rig
he e delle colonne (= dimensioni
 del quadro risultante finale)"
1070 PRINT " - coordinate (PRINT
 AT) per la centratura desiderat
a della stampa del quadro, indic
andone l'an-golo superiore sinis
tro"
1080 PRINT CHR$ 13;CHR$ 13;CHR$
13;"Regole per la composizione g
rafica:";CHR$ 13;" - immaginare
il quadro risultante come una ma
trice avente le righe come ordi
nata (1-";CHR$ 8; OVER 1;":22 a
lto->basso) e le lettere ""user
graphics"" come ascissa (a-";CHR
$ 8; OVER 1;":u sinistra->destra
); esempio di quadro 3x7:"
1090 PRINT CHR$ 13;"      a b c d
 e f g";CHR$ 13;" --------------
----";CHR$ 13;" 1        #    # #"
;CHR$ 13;" 2   #    # # #";CHR$ 1
3;" 3  # # # # # # #"
1100 PRINT CHR$ 13;" - per ogni
riga (nell'ordine 1-";CHR$ 8; OV
ER 1;":21) occorre creare tutti
i simboli grafici relativi alle
lettereinteressate e solo allora
 memo- rizzare la riga. Le case
lle vuote possono essere ignorat
e. Nell'esempio creare ""c e f"
" e memo-rizzare la riga, poi ""
```

Seguito listato 6.

```
a c d e"" ememorizzare la second
a riga, quindi ""a b c d e f g""
 e memorizza-re la terza riga."
1110 PRINT " - in ogni momento,
al completa-mento di una riga, e
 dopo averlamemorizzata, e' poss
ibile ottenere la stampa della c
omposizione risultante e prosegu
ire poi con la creazione delle r
ighe succes-sive"
1120 PRINT CHR$ 13;CHR$ 13;CHR$
13;"In caso di ""break"" inciden
tale, dovuto a scorrette immissi
oni o errori di SAVE/VERIFY, si
puo' riprendere correttamente l
'esecuzione eseguendo un ""GO TO
 520"".";CHR$ 13;CHR$ 13;"▒ No
n usare ""RUN"": cio' compor-ter
ebbe la perdita di quanto giafat
to! ▒"
1125 PRINT CHR$ 13
1130 GO TO 1001
7401 REM   © G.M.MELLINA ©
```

Listato 7 - COPY *screen composizione ottenuta con* GRAFCOMP.

Regressione e alta risoluzione grafica

Questo programma è solo una parte di un altro programma più vasto, scritto per VIC 20 e Apple III, per il calcolo di vari test statistici.

di **M. Mauro** e **F. Passariello**

La versione che qui presentiamo gira su VIC con espansione da 16 Kbyte e risolve il problema della regressione lineare: in parole più semplici traccia, attraverso una nuvola di punti, la retta che meglio si adatta ad essi. In pratica questo comporta il calcolo dell'equazione della retta che rende minima la somma dei quadrati delle distanze dei punti da essa (metodo dei minimi quadrati). Questo problema costituisce una occasione ideale per spendere qualche parola su alcuni espedienti di programmazione, che rendono piacevole l'uso del VIC.

Come si usa

Dopo aver caricato il programma Regressione se si adopera il registratore è necessario lasciare abbassato il tasto PLAY, altrimenti non sarà possibile ottenere una corretta esecuzione. Soddisfatti i preliminari, va dato il comando RUN, che provoca l'apparizione sullo schermo del messaggio ATTENDERE PREGO. L'intervallo sarà di circa 15 secondi per il trasferimento del set dei caratteri standard, più il tempo necessario per il caricamento del programma BIT.

Quest'ultimo tempo varia, ovviamente, a seconda che si adoperi il nastro (2 minuti) o il dischetto (15 secondi). Al termine della fase di attesa saranno visualizzati i nomi degli autori e poco dopo, la prima opzione che consente di ottenere l'output su video o su stampante. Se quest'ultima è l'opzione selezionata, sarà richiesto il numero delle cifre intere e di quelle decimali con cui si desidera vengano stampati i risultati. Se, invece, l'emissione avverrà sullo schermo, il numero delle cifre sarà standard. Altra differenza tra output su stampante e quello su video è determinata dal fatto che il prospetto stampato conterrà anche i dati in input, mentre i risultati su video saranno costituiti da una serie di videate, cui si accede pigiando un tasto qualunque. I dati in ingresso possono essere immessi da tastiera o, ancora, essere elaborati come numeri random. In ambedue i casi l'utente dovrà specificare su quante coppie di dati intende applicare il test. Se si è scelto di adoperare dei numeri casuali il passo successivo sarà l'output dei risultati, altrimenti si entrerà in una routine di immissione dati che funziona nel seguente modo: saranno richiesti tutti i dati della prima variabile e, successivamente, quelli della seconda (max. 200 coppie).

Possono essere immessi, al massimo, 8 numeri per volta costituiti da non più di 12 cifre, compreso il punto decimale. Terminato l'input di ogni gruppo di numeri, c'è la possibilità di effettuare delle correzioni, pigiando il tasto S. In alto a sinistra, in corrispondenza del primo numero, appare una freccia che può essere manovrata dall'alto verso il basso (premendo il tasto CRSR) o viceversa (SHIFT CRSR), finchè non ci si posizioni sul numero che si desidera modificare. La correzione si effettua premendo il tasto M e battendo il nuovo numero.

Terminate tutte le modifiche si procede pigiando il tasto F. Appena completata l'immissione il programma provvede al calcolo della media e della devianza dei due gruppi, della covarianza e del coefficiente di correlazione, dei coefficienti A e B, della retta di regressione, della devianza, varianza e deviazione standard campionaria dalla regressione e, infine, della deviazione standard campionaria di B, del T TEST e dei gradi di libertà. Il passo successivo consiste nella rappresentazione grafica cui, tuttavia, si può anche rinunciare. Per ottenere il diagramma di dispersione bisogna attribuire un nome all'ascissa e all'ordinata. Subito dopo, lo schermo assume il colore azzurro dello sfondo e dopo il veloce disegno delle coordinate visualizza il lento avanzare della retta di regressione. Il diagramma è completato dall'apparizione dei punti di dispersione e dai nomi dell'asse Y e dell'asse X. Le ultime due righe contengono i limiti dell'ascissa e dell'ordinata, generati come valori di default dall'algoritmo contenuto nel programma in base ai dati in input. Se uno di questi limiti supera, in valore assoluto, il numero 9999.99, allora sarà sostituito dal messaggio OVER LIM. Per uscire dallo schermo grafico è sufficiente pigiare un tasto qualunque. Fatto ciò si può ottenere una rappresentazione grafica più compatta scegliendo nuovi limiti per i due assi, in modo da escludere zone di scarso interesse.

L'ultima opzione consente di porre termine all'esecuzione del programma o di ripetere il test, con l'immissione di nuovi dati.

La grafica

L'output grafico costituisce una caratteristica interessante di questo programma. La finestra grafica coincide con l'intero schermo del VIC (22x23 caratteri) e non è persa la possibilità di inserire commenti, posti in qualsiasi posizione del video. Personalmente, abbiamo scelto di limitare i commenti alle zone laterale sinistra e inferiore del grafico, ma in teoria non vi è alcun limite, potendo i caratteri essere anche sovrapposti al grafico. Come funziona? Tutti coloro che usano il VIC sanno che è possibile trasferire la mappa dei caratteri dalla ROM alla RAM, potendosi così ride-

SIMBOLI	TASTI
1 . . "[illegible]"	SHIFT CLR
2 . . "[illegible]"	CLR
3 . . "[illegible]"	SHIFT CRSR/UP
4 . . "[illegible]"	CRSR/DOWN
5 . "[illegible]"	CRSR/RIGHT
6 . . "[illegible]"	SHIFT CRSR/LEFT
7 . . "[illegible]"	CTRL RVS/ON
8 . . "[illegible]"	CTRL RVS/OFF
9 . . "[illegible]"	CTRL 2
10 . "[illegible]"	CTRL 7
11 . "[illegible]"	SHIFT INST + DEL

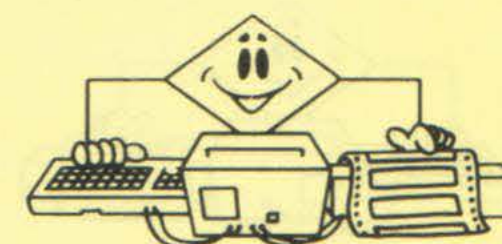



Figura 1 - Prospetto dei simboli grafici.

```
REGRESSIONE

VERSIONE CASSETTA

10 PRINT"[illegible]ATTENDERE PREGO":PRINT"[illegible]";
20 PZ=5120:PU=PZ:PRINT"[illegible]"
30 FORI=0TO2:POKE7168+I,0:NEXT:FORK=0TO1023:POKEPU+1024+K,PEEK(34816+K):NEXT
40 PRINT"POKE43,1:POKE44,28:POKE45,3:POKE46,28:POKE47,3:POKE48,28:POKE49,1:POKE5
0,2:CLR:NEW"
50 PRINT"[illegible]LOAD"CHR$(34)"BIT"CHR$(34)
60 PRINT"[illegible]RUN"
70 PRINT"[illegible]":POKE198,3:POKE631,13:POKE632,13:POKE633,13:END

MODIFICHE PER
FLOPPY DISK

50 PRINT"[illegible]LOAD"CHR$(34)"BIT"CHR$(34)",8"
60 PRINT"[illegible]RUN"
70 PRINT"[illegible]":POKE198,3:POKE631,13:POKE632,13:POKE633,13:END
```

Listato 1 - Il programma Regressione.

finire ogni carattere per produrre opportuni disegni o diagrammi. In genere basta far corrispondere ad ogni carattere così ridefinito una ed una sola locazione della mappa video. In tal modo, però, essendo 256 i caratteri ASCII (normal + reverse), si può usufruire di una grafica di soli 256x8x8 = 16384 pixel. Se si adopera l'opzione per la definizione di caratteri 8x16, bisogna però ridurne anche il numero, per non superare il massimo numero di punti memorizzabili nella mappa del video (506 locazioni video x 8 byte x 8 bit). Il nostro programma invece utilizza un altro metodo: non vi è corrispondenza biunivoca tra mappa caratteri e mappa video, in quanto è facile vedere come in tutti i grafici la maggior parte dello schermo è occupata dal carattere bianco. Il programma utilizza così solo 128 caratteri per la grafica (sono molti!), in quanto il campo reverse è riempito con la mappa dei caratteri minuscoli e normal, utilizzati per i commenti al video. Il corpo del programma di grafica è contenuto nelle routine 2140-2190 e 2200-2220. Il numero di pixel ottenibile è 32384 (22x23 caratteri).

Perchè due programmi

Quando si fa girare il programma, nella fase iniziale bisogna attendere un tempo variabile da circa 30 secondi quando si usa il floppy disk a circa 2 minuti e 20 secondi, se si è equipaggiati di solo cassette-recorder. Sullo schermo appare un laconico "attendere prego". In realtà, in quei momenti avvengono cose molto importanti. Adoperando l'espansione da 16 Kbyte (lo stesso anche per la 8 Kbyte), la mappa caratteri deve essere trasferita all'inizio dell'area di memoria riservata al programma, subito dopo la mappa video.

Questo trasferimento non può avvenire all'interno di un programma, perchè dovendosi cancellare tale area per pulire lo schermo, verrebbe ad essere cancellato anche parte del programma in esecuzione. Inoltre è necessario spostare tutti i puntatori di inizio programma, variabili e variabili indicate. Quando si adopera una espansione 3 Kbyte o si è nella configurazione standard, la mappa va spostata alla fine della memoria e quindi deve essere spostato in basso il puntatore di fine area programma e di inizio area stringhe (le stringhe sono memorizzate a scendere). Nel nostro caso, il tutto è ottenuto con un programma introduttivo (Regressione), talmente corto che non fuoriesce dalla zona di memoria 4609-5120, il quale memorizza i caratteri minuscoli in campo diretto nella zona del campo reverse (da 6148 in poi, per 1 Kbyte), mentre l'azzeramento dei caratteri è lasciato al programma successivo (BIT), nelle istruzioni alle righe 660-670. Il primo programma scrive sullo schermo dei comandi invisibili (in bianco) e termina, forzato con il solito espediente (**Bit** n° 39, pag. 118) ad eseguire per tre volte l'istruzione RETURN. L'esecuzione che scatta dopo

il comando END comporta lo spostamento dei puntatori, fuori programma, quindi il caricamento del programma successivo (BIT) e la sua esecuzione. Attenti al numero esatto di simboli grafici e a mantenere pigiato il tasto PLAY, perchè principalmente da questi accorgimenti dipende l'intero automatismo. Se per curiosità volete osservate l'espediente, cancellate la linea 10 del programma Regressione e le varie fasi saranno visualizzate sullo schermo.

Attenzione! Si tratta di due programmi separati, di cui il secondo non può essere eseguito se prima non si è fatto girare almeno una volta il programma introduttivo.

Questi programmi devono essere registrati in successione, se posti su cassetta, in quanto Regressione richiama automaticamente BIT e lo manda in esecuzione. Se si usa il floppy disk, basta apportare le opportune modifiche, così come si vede nel listato.

Routine

110-220 I numeri sono restituiti secondo il formato desiderato, in base alle cifre intere e decimali richieste.

300-360/370-400 Necessarie per il calcolo automatico dei limiti del diagramma.

470-490 Rende più agevole il posizionamento dei messaggi sullo schermo. Con questa routine si sono volute imitare le funzioni delle istruzioni VTAB-HTAB dell'Apple II e VPOS e HPOS dell'Apple III. Le variabili RG e CL individuano, rispettivamente, la riga e la colonna da cui si vuole far partire la visualizzazione di un messaggio. I due valori costituiscono l'input della routine, il cui output è determinato dai valori che vanno inseriti nelle locazioni 209 e 210, contenenti l'indirizzo della mappa dello schermo corrispondente alla posizione del cursore.

Terminata l'esecuzione della routine, è possibile scrivere un messaggio nel punto desiderato (vedi righe 730-740).

500-520 Consentono la preparazione dello schermo prima di ciascuna videata.

530-570 Visualizzano, sull'ultima riga del video, il messaggio SCEGLI UN NUMERO. Tale messaggio lampeggia finchè non sia stato scelto un numero compreso tra quelli indicati.

680-1200 Le parti più interessanti di questa routine riguardano l'introduzione (linee 880-950) e la modifica (960-1200) dei dati. L'input avviene per gruppi di otto numeri.

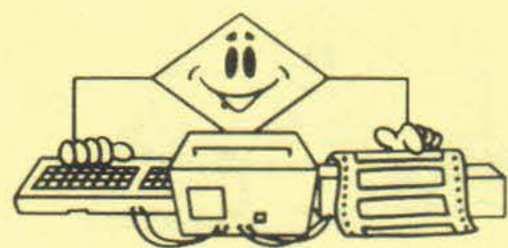
Quando la variabile E è uguale a 9 o termina l'immissione, si passa alla fase della modifica. Da notare come sia stato necessario discriminare il primo dato immesso rispetto ai successivi sette, in quanto il VIC considera ciò che viene scritto in quel caso come un tutt'uno con quanto è presente nella riga precedente del video. Sono nell'input del primo dato si è dovuto, perciò, ricorrere ad un artificio, come appare nella riga 930. Molto funzionale si dimostra il sistema adottato per la modifica dei dati. Infatti è sufficiente spostare una freccia posizionata alla sinistra dei numeri. Affinchè il programma possa sempre conoscere la posizione in cui si trova la freccia è stato utilizzato il contatore FF, che è incrementato o decrementato a seconda che la freccia scenda o salga. Una volta richiesta la modifica, appare un asterisco, mentre il numero da modificare scompare (riga 1180) in seguito al PRINT di una serie di 20 cursori a destra e 14 DEL. L'operazione di modifica può essere effettuata quante volte si vuole. Una curiosità relativa sempre alla linea 1180: scrivete questa riga così come si trova.

Chiedetene, poi il LIST sullo schermo. È inutile che vi affanniate a ribatterla, nel timore di non avere digitato bene. I caratteri grafici che non compaiono sul video sono, invece, ben visibili nel listato della stampante.

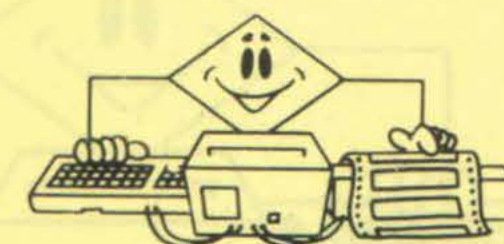
Listato 2 - Il programma BIT.

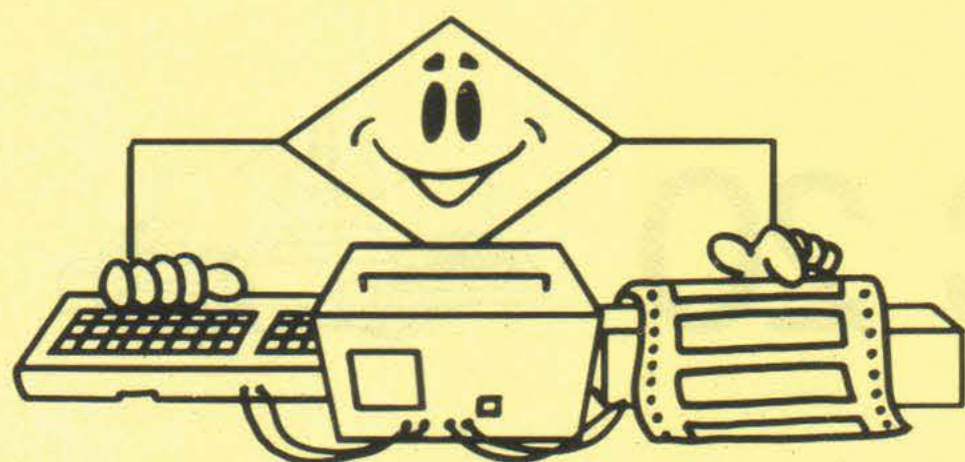
```
          B I T
10 PRINT"▒▒";:O=80:C$="                                    "
20 PRINT"▒";:WH=PEEK(210)*256+PEEK(209):WH$="======================"
30 DIMX(200,2)
40 GOSUB500:PRINT"▒▒▒▒▒▒▒▒▒▒▒▒REGRESSIONE":PRINT"▒▒▒▒▒▒▒PROGRAMMA DI"
50 PRINT"▒▒▒▒MICHELE  MAURO":PRINT"▒▒▒FAUSTO PASSARIELLO"
60 GOSUB660:GOTO1210
70 REM    ATTESA
80 PRINT"▒▒▒▒▒▒PREMI UN ▒TASTO":PRINT"▒▒▒▒PER CONTINUARE"
90 GETA$:IFA$=""THEN90
100 PRINT"▒":RETURN
110 REM    FORMATO
120 IC=CI:DC=CD
130 BS=SGN(B):BB=ABS(B):BI=INT(BB):BF=BB-BI:BI$=STR$(BI)
140 BF$=STR$(INT(BF*10↑CD)):BF$=RIGHT$(BF$,LEN(BF$)-1)
150 XW=CD-LEN(BF$):IFXW=0THEN170
160 FORZK=1TOXW:BF$="0"+BF$:NEXT
170 BL=LEN(BI$):D$=" ":IFBS=-1THEND$="-"
180 IFCI<BLTHENCI=BL
190 T$=".":IFBF<1E-6THENT$=" ":BF$=LEFT$(C$,CD)
200 IFCD=0THENT$="":BF$=""
210 B$=LEFT$(C$,CI+1-BL)+D$+RIGHT$(BI$,BL-1)+T$+BF$
220 CI=IC:CD=DC:RETURN
230 REM
240 GOSUB500:PRINT"▒▒▒▒PARAMETRI STAMPA"
250 PRINT"▒▒▒▒▒CIFRE INTERE******":PRINT"▒(MAX.10)";
260 PRINTTAB(15);"";:INPUTCI:IFCI<0ORCI>10THENPRINT"▒";:GOTO260
270 PRINT"▒▒▒▒▒CIFRE DECIMALI****":PRINT"▒(MAX.10)";
280 PRINTTAB(15);"";:INPUTCD:IFCD<0ORCD>10THENPRINT"▒";:GOTO280
290 RETURN
```

Seguito programma BIT.

```
300 REM    MINIMO
310 I=-10:K=0:IFZ9<0THENZ9=-Z9:K=1
320 IFZ9>10↑ITHENI=I+1:GOTO320
330 X9=10↑(I-1):HT=2
340 IFZ9>X9*HTTHENHT=HT+1:GOTO340
350 Z9=X9*(HT-1+K):IFKTHENZ9=-Z9
360 RETURN
370 REM    RANGE
380 Z6=Z9
390 Z8=Z7-Z6:IFZ8>.2*Z5THENZ9=Z8:GOSUB300:Z6=Z6+Z9:GOTO390
400 Z7=Z6:RETURN
410 REM  VIDEO/STAMPA
420 GOSUB500:PRINT"████████▌▌VUOI I RISULTATI":PRINT"██▌SU VIDEO O STAMPANTE"
430 PRINT"█▌▌▌▌▌▌▌▌▌▌▌█V/S"
440 GETEL$:IFEL$="V"THENCI=9:CD=9:RETURN
450 IFEL$="S"THENGOSUB230:RETURN
460 GOTO440
470 REM   CURSORE
480 A=WH+(RG-1)*22+CL-1:RH=INT(A/256):CH=A-RH*256:POKE210,RH:POKE209,CH
490 RETURN
500 REM FORMATO VIDEO
510 PRINT"□":RG=3:CL=1:GOSUB470:PRINTWH$:RG=21:CL=1:GOSUB470:PRINTWH$+"█";
520 RETURN
530 REM   OPZIONE
540 CC=0:RG=22:CL=4
550 IFCC=1THENCC=CC+1:PRINT"■";:GOTO570
560 PRINT"█";:CC=1
570 GOSUB470:PRINT"SCEGLI UN NUMERO▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌▌";:FORI=1TO500:NEXT:IFCC=2THE
NRETURN
580 GOTO550
590 REM VARIABILI
600 GOSUB500:PRINT"█"+Z9$
610 REM COPPIE
620 GOSUB500:PRINT"█"+Z9$
630 PRINT"██████▌▌▌NUMERO COPPIE":PRINT"██▌(MIN. 3 - MAX. 200)██"
640 INPUT"▌▌▌▌▌▌▌▌▌▌▌";S:IFS<3ORS>200THENPRINT"□";:GOTO640
650 FORI=1TON:S(I)=S:NEXT:RETURN
660 REM AZZERA CARATTERI
670 PZ=5120:PU=PZ:FORK=0TO1023:POKEPZ+K,0:NEXT:RETURN
680 REM INTRODUZIONE
690 REM     DATI
700 GOSUB410
710 Z9$="▌▌█INTRODUZIONE DATI"
720 GOSUB500:PRINT"██▌"+Z9$
730 RG=10:CL=1:GOSUB470
740 PRINT"█1■....INPUT DA TASTIERA"
750 RG=15:GOSUB470:PRINT"█2■.......NUMERI CASUALI"
760 RG=22:CL=4
770 GOSUB530:GETA$:IFVAL(A$)<1ORVAL(A$)>2THEN770
780 ONVAL(A$)GOTO820,790
790 REM
800 GOSUB610
810 RU=RND(-TI):FORJ=1TON:FORI=1TOS(J):X(I,J)=RND(1):NEXTI,J:RETURN
820 REM NUOVI DATI
830 GOSUB610
840 GOSUB860:RETURN
850 RG=4:CL=10:GOSUB470:PRINT"█SERIE";T9;"■":RETURN
860 H7=0:FORT9=1TON:GOSUB500:RG=2:CL=1:GOSUB470:PRINT"█";Z9$
870 GOSUB850:E=0:GOSUB 880:NEXT T9:RETURN
880 FORI=1TOS(T9)
890 E=E+1
900 IFE=9THENH7=1:GOSUB960:GOSUB500:RG=2:CL=1:GOSUB470:PRINT"█";Z9$:GOSUB850:E=1
910 RG=E*2+3:CL=4:GOSUB470
920 ZZ$=MID$(STR$(I),2,LEN(STR$(I))-1):PRINT"█";ZZ$;")■":PRINT"□▌▌▌▌▌▌▌▌";
930 INPUTA$:IFE=1THENA$=MID$(A$,23,13)
940 X(I,T9)=VAL(A$):PRINT"█";
950 NEXT I:H7=0:GOSUB 960:PRINT"□";: RETURN
960 RG=22:CL=3:GOSUB470:PRINT"█MODIFICA DATI(S/N)"
970 GET A$
980 IF A$="N" THEN E=1:RETURN
990 IFA$="S" THENFF=1:RG=5:CL=1:GOSUB470:PRINT">":GOTO 1010
1000 GOTO 970
1010 RG=22:CL=2:GOSUB470:PRINT"█MODIFICA=M    FINE=F□":GOSUB1140
```

VIC 20

Seguito listato programma BIT.

```
1020 GET A$
1030 IF A$="▒"THENFF=FF+1:GG=1:GOTO1080
1040 IF A$="▔"THEN FF=FF-1:GG=-1:GOTO1080
1050 IF A$="M"THEN1180
1060 IFA$="F"THENE=1:RETURN
1070 GOTO 1020
1080 IF FF<1 THENFF=1:GOTO 1020
1090 IF FF>EETHENFF=EE:GOTO 1020
1100 PRINT" " :GOSUB 1120
1110 GOTO 1020
1120 IF GG=1 THEN PRINT"▒▷▔":RETURN
1130 PRINT"▔▔▔▷▌";:RETURN
1140 EE=E:IF E=9 THEN EE=8
1150 FOR TT=1 TO 2*EE+2
1160 PRINT"▔";
1170 NEXT TT:RETURN
1180 PRINT"*████████████████████████████████████████";:INPUTA$:IFFF=1THENA$=MID$(A$,23
,13)
1190 X(I-1-E+H7+FF,T9)=VAL(A$)
1200 PRINT"▔▷▌";:GOTO1020
1210 REM CORRELAZIONE
1220 N=2:GOSUB680
1230 REM   CALCOLO
1240 WZ=0
1250 FORI=1TOS:FORJ=1TO2:AS(J)=AS(J)+X(I,J):Q(J)=Q(J)+X(I,J)↑2:NEXTJ
1260 XY=XY+X(I,1)*X(I,2):NEXTI
1270 FORI=1TO2:M(I)=AS(I)/S:S(I)=Q(I)-S*M(I)↑2:NEXT
1280 COV=XY-AS(1)*AS(2)/S
1290 IFS(1)<1E-6ORS(2)<1E-6THENWZ=1:GOTO1310
1300 R=COV/SQR(S(1)*S(2))
1310 F=S-2:IFS(1)<E-6THENWZ=2:GOTO1370
1320 BV=COV/S(1):AV=M(2)-BV*M(1)
1330 DXY=S(2)-COV↑2/S(1)
1340 S2=DXY/F:SV=S2↑.5:SB=SV/S(1)↑.5
1350 IFSB<1E-6THENWZ=3:GOTO1370
1360 T=BV/SB
1370 REM STAMPA
1380 IFEL$="V"THENOO$=CHR$(13):PRINT"▔";:GOTO1410
1390 OPEN4,4:CMD4,"CORRELAZIONE E REGRESSIONE LINEARE":PRINT
1400 FORI=1TOS:FORJ=1TO2:PRINTX(I,J);:NEXT:PRINT:NEXT:PRINT
1410 FORI=1TO2:PRINT"GRUPPO";I:PRINT:PRINT"MEDIA";OO$;:B=M(I):GOSUB110:PRINTB$
1420 PRINT"DEVIANZA";OO$;:B=S(I):GOSUB110:PRINTB$:PRINT:NEXT
1430 PRINT"COVARIANZA";OO$;:B=COV:GOSUB110:PRINTB$
1440 IFEL$="V"THENGOSUB70
1450 IFWZTHENPRINT"COEFFICIENTE DI CORRELAZIONE NON CALCOLABILE":GOTO1470
1460 PRINT"COEFFICIENTE DI";OO$;"CORRELAZIONE";OO$;:B=R:GOSUB110:PRINTB$;OO$
1470 IFWZ=2THENPRINT"PARAMETRI DELLA REGRESSIONE NON CALCOLABILI":GOTO1590
1480 PRINT"COEFFICIENTE B DELLA";OO$;"RETTA DI REGRESSIONE";OO$;:B=BV:GOSUB110:P
RINTB$;OO$
1490 PRINT"COEFFICENTE A DELLA";OO$;"RETTA DI REGRESSIONE";OO$;:B=AV:GOSUB110:PR
INTB$
1500 IFEL$="V"THENGOSUB70
1510 PRINT"DEVIANZA DALLA "OO$"REGRESSIONE"OO$;:B=DXY:GOSUB110:PRINTB$
1520 PRINT"VARIANZA DALLA "OO$"REGRESSIONE"OO$;:B=S2:GOSUB110:PRINTB$
1530 PRINT"DEVIAZIONE STANDARD "OO$"CAMPIONARIA DALLA "OO$"REGRESSIONE"OO$;:B=SV
:GOSUB110:PRINTB$
1540 IFEL$="V"THENGOSUB70
1550 PRINT"DEVIAZIONE STANDARD "OO$"CAMPIONARIA DI B"OO$;:B=SB:GOSUB110:PRINTB$
1560 IFWZ=3THENPRINT"T TEST NON CALCOLABILE":GOTO1580
1570 PRINT"T TEST"OO$;:B=T:GOSUB110:PRINTB$
1580 PRINT"GRADI DI LIBERTA'";F
1590 IFEL$="S"THENPRINT#4:CLOSE4:GOTO1610
1600 GOSUB70
1610 GOSUB500:PRINT"▒▒▒▒▒▒▒██VUOI LA GRAFICA ?▒███▌S/N"
1620 GETA$:IFA$<>"S"ANDA$<>"N"THEN1620
1630 IFA$="S"THEN1680
1640 GOSUB500:PRINT"▒▒▒▒▒▒▒████VUOI FINIRE ?▒███▌S/N"
1650 GETA$:IFA$<>"S"ANDA$<>"N"THEN1650
1660 IFA$="S"THENPRINT"▔":END
1670 RUN
1680 REM GRAFICA
1690 GOSUB500:Z9$="▒██████▌GRAFICA":PRINTZ9$:INPUT"▒▒▒▒██LABEL X";X$:INPUT"▒▒▌
█LABEL Y";Y$
```

Seguito listato programma BIT.

```
1700 X0=X(1,1):X1=X(1,1):Y0=X(1,2):Y1=X(1,2)
1710 FORJ=2TOS:IFX(J,1)<X0THENX0=X(J,1)
1720 IFX(J,1)>X1THENX1=X(J,1)
1730 IFX(J,2)<Y0THENY0=X(J,2)
1740 IFX(J,2)>Y1THENY1=X(J,2)
1750 NEXT
1760 RX=X1-X0:RY=Y1-Y0
1770 X0=X0-.1*RX:X1=X1+.1*RX:Y0=Y0-.1*RY:Y1=Y1+.1*RY
1780 DX=X1-X0:DY=Y1-Y0
1790 SX=DX/6:SY=DY/6:Z9=SX:GOSUB300:SX=Z9:Z9=SY:GOSUB300:SY=Z9
1800 Z9=X0:GOSUB300:Z7=X0:Z5=DX:GOSUB370:X0=Z7:Z9=Y0:GOSUB300:Z7=Y0:Z5=DY:GOSUB3
70
1810 DX=X1-X0:DY=Y1-Y0
1820 REM MM=DENSITA',RR=RIGA,CC=COLONNA DI X0,Y0
1830 BC=37888:RR=5:CC=3:MM=1
1840 PRINT"███";:FORT=1TO23-RR:PRINTTAB(CC-1);"█|":NEXT
1850 PRINTTAB(CC);:FORT=1TO22-CC:PRINT"█‾";:NEXT
1860 PR=(22-CC)*8-1:PC=(23-RR)*8-1
1870 POKE36869,205:POKE36879,110
1880 FORXP=X0TOX1STEPDX/PC/MM:YP=AV+BV*XP:IFYP<Y0ORYP>Y1THEN1900
1890 GOSUB2140
1900 NEXTXP
1910 FORI=1TOS:XP=X(I,1):YP=X(I,2)
1920 IF(XP<X0)+(XP>X1)+(YP<Y0)+(YP>Y1)THEN1940
1930 GOSUB2140
1940 NEXT:PRINT"██";
1950 FORI=1TOLEN(Y$):PRINT"█ "MID$(Y$,I,1):IFLEN(Y$)>8THEN1970
1960 PRINT" "
1970 NEXT:OQ$=" ":OQ=2:IFLEN(X$)>8THENOQ$="":OQ=1
1980 PRINT"██████████████████████";:PRINTSPC(22-OQ*LEN(X$));:IFX$=""THENOQ$=""
1990 FORI=1TOLEN(X$):PRINT"█"MID$(X$,I,1)+OQ$;:NEXT
2000 PRINT"█ =====================";:CI=4:CD=2
2010 N9$="OVER LIM"
2020 PRINT"█";:B=X0:GOSUB110
2030 IFLEN(BI$)-1>CITHENB$=N9$
2040 PRINTB$;"<=X<=";:B=X1:GOSUB110
2050 IFLEN(BI$)-1>CITHENB$=N9$
2060 PRINTB$" ";
2070 PRINT"█";:B=Y0:GOSUB110
2080 IFLEN(BI$)-1>CITHENB$=N9$
2090 PRINTB$;"<=Y<=";:B=Y1:GOSUB110
2100 IFLEN(BI$)-1>CITHENB$=N9$
2110 PRINTB$;
2120 GETA$:IFA$=""THEN2120
2130 PRINT"██":POKE36869,192:POKE36879,27:GOTO2260
2140 X=(XP-X0)*PR/DX:Y=(YP-Y0)*PC/DY
2150 A%=X/8:B%=Y/8:R1%=X-A%*8:R2%=Y-B%*8:V%=2↑(7-R1%):C%=-(B%+RR)*22+A%+CC
2160 IFKTHENK=0:GOTO2190
2170 IFC%<>C1THENGOSUB2200
2180 IF(PZ-PU)/8=32THENPZ=PZ+8
2190 C1=C%:PB=PZ+7-R2%:POKEPB,V%ORPEEK(PB):RETURN
2200 ZQ=PEEK(WH+484+C1):IFZQ=32THEN2240
2210 IFZQ>127THEN2230
2220 FORKQ=0TO7:POKEPU+ZQ*8+KQ,PEEK(PZ+KQ)ORPEEK(PU+ZQ*8+KQ):POKEPZ+KQ,0:NEXT:RE
TURN
2230 FORKQ=0TO7:POKEPZ+KQ,PEEK(PZ+KQ)ORPEEK(PU+ZQ*8+KQ):NEXT
2240 POKEWH+484+C1,(PZ-PU)/8:POKEBC+484+C1,1:PZ=PZ+8
2250 RETURN
2260 GOSUB500:Z9$="█GRAFICA"
2270 PRINT"████████"Z9$:PRINT"██████████VUOI UNA SCALA":PRINT"██████DIFFERENTE ?█
█████S/N"
2280 GETA$:IFA$<>"S"ANDA$<>"N"THEN2280
2290 IFA$="N"THEN1640
2300 GOSUB500:Z9$="██████LIMITI GRAFICO":PRINTZ9$:PRINT"███"
2310 INPUT"X MIN.█████████";X0
2320 INPUT"██ MAX.";X1
2330 IFX1-X0<=1E-6THENPRINT"███";:GOTO2310
2340 PRINT
2350 INPUT"Y MIN.";Y0
2360 INPUT"██ MAX.";Y1
2370 IFY1-Y0<=1E-6THENPRINT"███";:GOTO2350
2380 GOSUB660:GOTO1780
2390 PRINT"█";:END
```

ZX 81

ZX micologo. Attenzione, comunque, agli avvelenamenti!

Con questo programma lo ZX81 è in grado di riconoscere una sessantina di funghi tra i più comuni, e sarà quindi un valido aiuto ed allo stesso tempo uno stimolo per coloro che vorranno avvicinarsi per la prima volta alla micologia.

di **A. Polano**

Naturalmente il programma non vuole e non può sostituire alcun testo valido che tratti dell'argomento e del quale, anzi, si consiglia la lettura per integrare ed arricchire il numero di informazioni date dal computer in special modo relativamente alla commestibilità dei funghi. Questa parte infatti è trattata soltanto nel caso degli unici tre funghi veramente mortali che si conoscono.

Una raccomandazione va fatta in merito alla caratteristica "sapore" del fungo in esame: il fungo si può assaggiare solo quando si è perfettamente sicuri della sua commestibilità; per poter individuare tale caratteristica ci si dovrà affidare per le prime volte, con buona approssimazione, al proprio olfatto.

E veniamo ora alla descrizione del programma. Con uno dei soliti cicli FOR-NEXT, che poi andranno cancellati, si inseriscono le due tabelle F$ ed N$ opportunamente dimensionate. Fatto ciò, si immette il programma come da listato e, a lavoro ultimato, lo si salva su nastro con il comando diretto GO TO 9998. A questo punto il programma va in autostart e si potranno verificare i seguenti tre casi: la percentuale di correlazione tra il fungo descritto ed il fungo in memoria è inferiore al cinquanta per cento ed allora il nostro ZX affermerà di non conoscerlo. Al contrario vi potranno essere due o più funghi che hanno le stesse probabilità di risultare quelli con i caratteri da noi impostati, la qual cosa verrà segnalata con il messaggio della linea 2120.

L'ultimo caso invece è quello in cui non vi sono dubbi e pertanto viene visualizzato il nome del fungo (in latino). Un'ultima avvertenza: per le descrizioni del fungo in esame o per eventuali ampliamenti delle tabelle memorizzate ci si potrà avvalere della tavola di identificazione.

```
ALCUNE VIDEATE DEL PROGRAMMA:

MICOLOGIA
                    PREMI NEW-LINE

IL FUNGO CHE PIU" SI AVVICINA
(87.5 %) ALLA TUA DESCRIZIONE
SI CHIAMA:

        AMANITA VERNA

        ANCORA? S/N
```

REMarks

1-100	Presentazione del programma.
110	Viene inizializzata K$ che conterrà tutte le caratteristiche del fungo in esame man mano richieste dal computer con le linee dalla 120 alla 950.
1000	Subroutine per la formazione di K$.
1500-1690	Vengono richieste due caratteristiche individuate con certezza in modo da abbreviare e selezionare la successiva ricerca.
2000-2110	Avviene il controllo tra il fungo descritto ed i funghi contenuti in memoria.
2113-3000	Output del programma.
F$	Tabella con i nomi dei funghi.
N$	Tabella con le caratteristiche dei funghi.
K$	Caratteri descritti dall'utente.
M ed N	Posizioni dei caratteri individuati con certezza.
R	Numero dei funghi individuati dal computer.
P	Percentuale di correlazione del fungo esaminato durante il ciclo di ricerca.
K	Percentuale di correlazione del fungo individuato.
J	Posizione in tabella del fungo individuato.

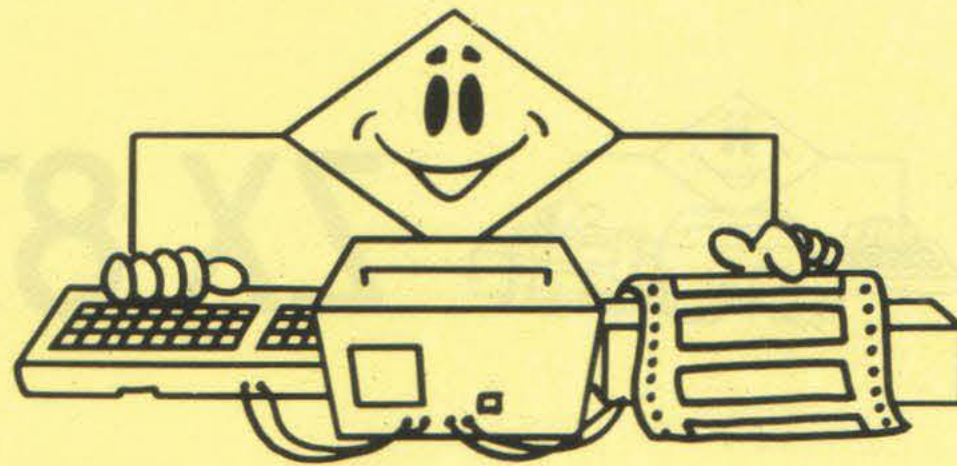
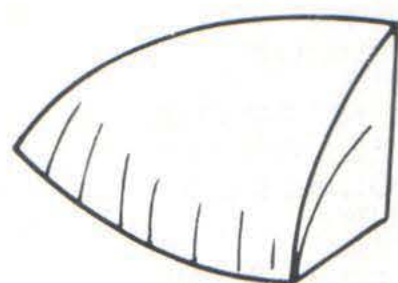
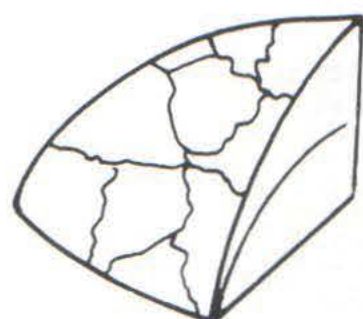
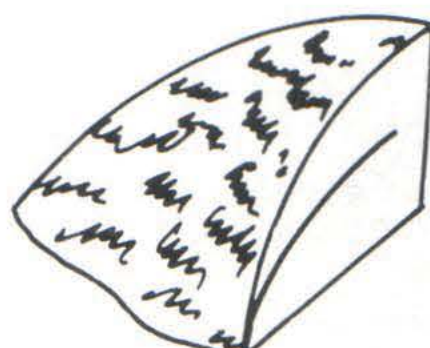
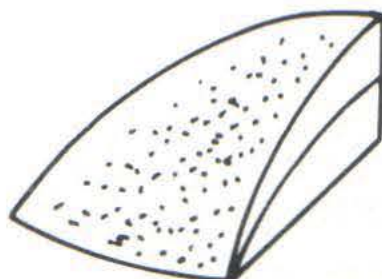
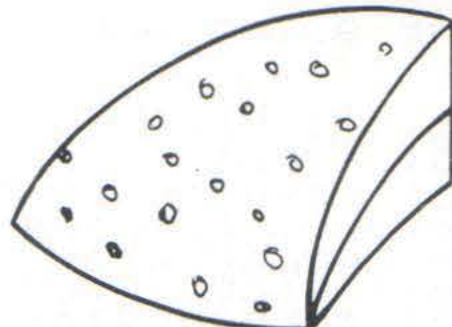
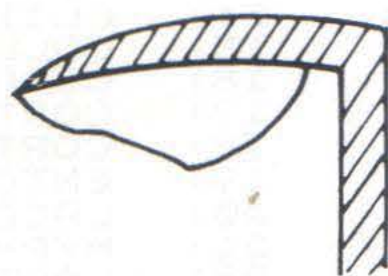
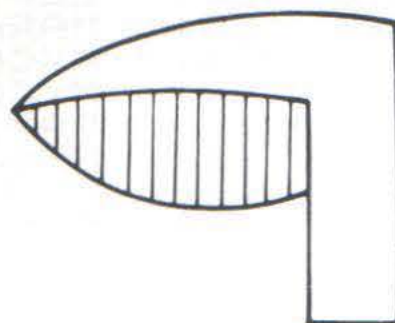
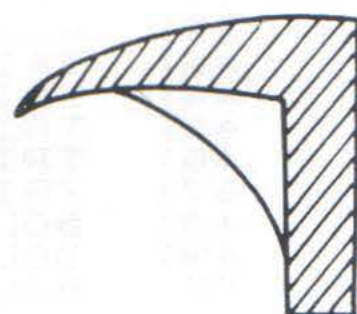
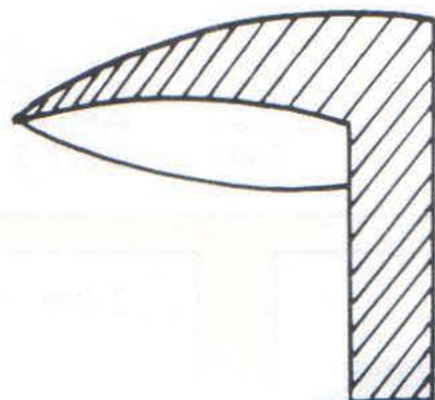



TAVOLA DI IDENTIFICAZIONE

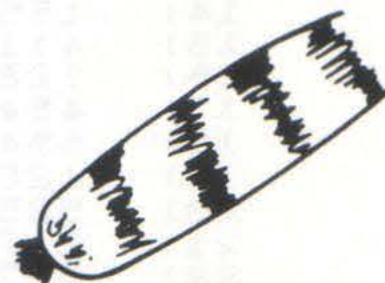
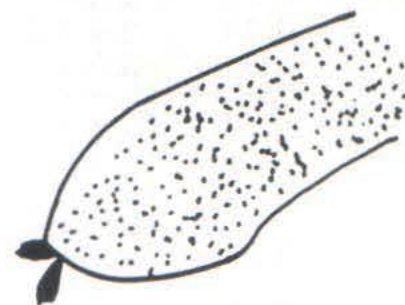
ORNAMENTAZIONI DEL CAPPELLO

CON MARGINE STRIATO

SCREPOLATO

DESQUAMATO

VELLUTATO

CON PLACCHE

CON VERRUCHE

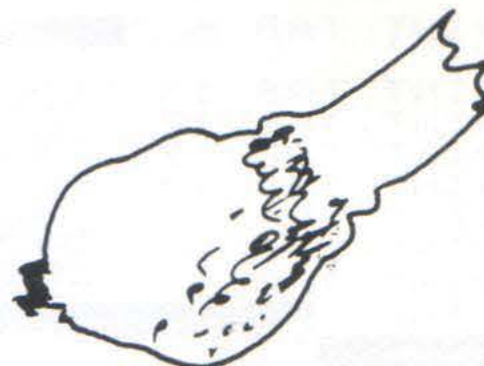
FIBRILLOSO

FORMA DELL'IMENIO

LAMELLE LIBERE

TUBULI

LAMELLE DECORRENTI

LAMELLE ADNATE

LAMELLE ANNESSE

ORNAMENTAZIONI DEL GAMBO

ZONATO

PUNTEGGIATO

RETICOLATO

VOLVA A SACCO

VOLVA NON A SACCO

ANELLO LISCIO E STRIATO

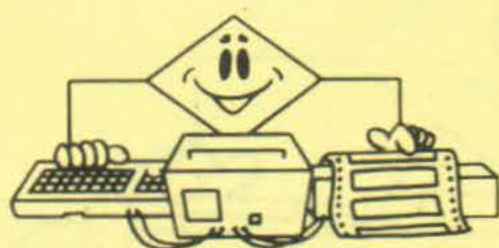
Figura 1 - La tavola di identificazione che permette l'esame del fungo ed anche l'ampliamento delle tabelle memorizzate.

ZX 81

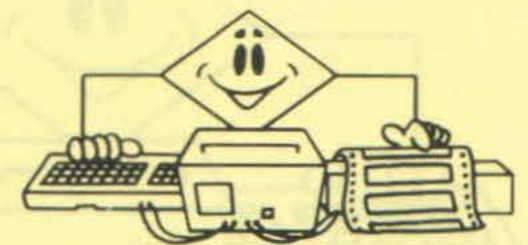
Figura 2 - Le due tabelle di nomi e caratteristiche da introdurre con un ciclo FOR...NEXT.

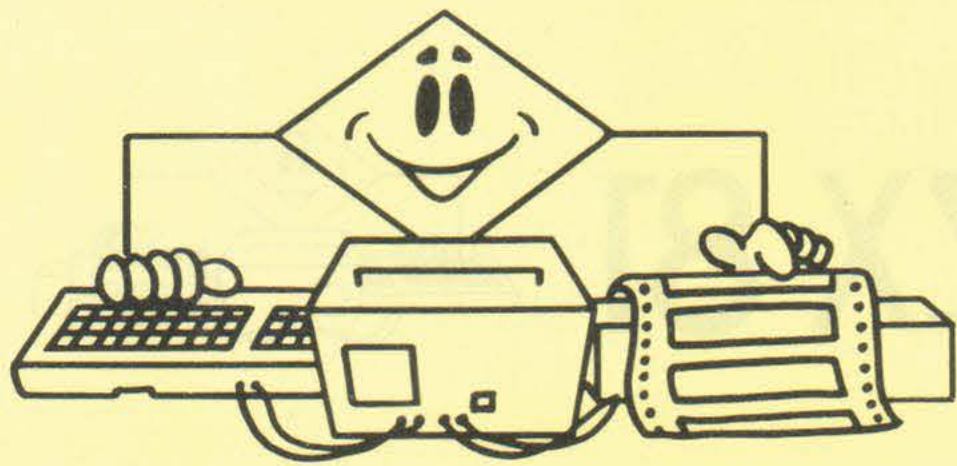
N.	Caratteristiche
1)	72117411
2)	12111414
3)	31212441
4)	31111611
5)	43111511
6)	44114631
7)	24111614
8)	23134711
9)	21262611
10)	81864881
11)	31141811
12)	11141813
13)	31243841
14)	15541512
15)	41134611
16)	12811811
17)	45434841
18)	65436881
19)	42511811
20)	52545881
21)	21722544
22)	35733865
23)	31233376
24)	15131875
25)	46111731
26)	36111711
27)	11511711
28)	11121811
29)	28122815
30)	58555814
31)	45244844
32)	11121713
33)	47224711
34)	61141811
35)	42511811
36)	11511731
37)	17511731
38)	16511744
39)	62131811
40)	81131815
41)	31131815
42)	77131811
43)	12121812
44)	11121812
45)	51525381
46)	12122811
47)	46121811
48)	51272151
49)	29271711
50)	25272811
51)	41171111
52)	39273741
53)	41574134
54)	35374151
55)	45374251
56)	47174211
57)	31174211
58)	41372151
59)	11111411
60)	00000000

N.	Nome
1)	AMANITA PHALLOIDES
2)	AMANITA VIROSA
3)	AMANITA CAESAREA
4)	AMANITA MUSCARIA
5)	AMANITA PANTHERINA
6)	AMANITA RUBESCENS
7)	AMANITA CITRINA
8)	ARMILLARIA MELLEA
9)	CANTHARELLUS CIBARIUS
10)	CANTHARELLUS CORNUCOPIOIDES
11)	CLITOCYBE INVERSA
12)	CLITOCYBE NEBULARIS
13)	CLITOCYBE OLEARIA
14)	CLITOPILUS PRUNULUS
15)	COLLYBIA FUSIPES
16)	COPRINUS ATRAMENTARIUS
17)	CORTINARIUS ORELLANUS
18)	CORTINARIUS VIOLACEUS
19)	ENTOLOMA LIVIDUM
20)	LACCARIA LACCATA
21)	HYPHOLOMA FASCICULARE
22)	LACTARIUS DELICIOSUS
23)	LACTARIUS VOLEMUS
24)	LACTARIUS PIPERATUS
25)	LEPIOTA PROCERA
26)	LEPIOTA CRISTATA
27)	LEPIOTA NAUCINA
28)	MARASMIUS OREADES
29)	MYCENA INCLINATA
30)	MYCENA PURA
31)	PAXILLUS INVOLUTUS
32)	PHOLIOTA AEGERITA
33)	PHOLIOTA MUTABILIS
34)	PLEUROTUS OSTREATUS
35)	PLUTEUS CERVINUS
36)	PSALLIOTA BISPORA
37)	PSALLIOTA CAMPESTRIS
38)	PSALLIOTA XANTHODERMA
39)	RUSSULA CYANOXANTHA
40)	RUSSULA NIGRICANS
41)	RUSSULA EMETICA
42)	RUSSULA VIRESCENS
43)	TRICHOLOMA COLUMBETTA
44)	TRICHOLOMA GEORGII
45)	TRICHOLOMA NUDUM
46)	TRICHOLOMA PORTENTOSUM
47)	TRICHOLOMA TERREUM
48)	BOLETUS REGIUS
49)	BOLETUS LUTEUS
50)	BOLETUS CAVIPES
51)	BOLETUS EDULIS
52)	BOLETUS ELEGANS
53)	BOLETUS FELLEUS
54)	BOLETUS PURPUREUS
55)	BOLETUS ERYTHROPUS
56)	BOLETUS SCABER
57)	BOLETUS RUFUS
58)	BOLETUS LURIDUS
59)	AMANITA VERNA
60)	DATI INSUFFICIENTI

Figura 3 - Il listato del programma ZX micologo: ma attenzione ad usarlo con le dovute precauzioni! Gli avvelenamenti da funghi sono molto comuni anche tra gli esperti.... di computer.

```
   1 SLOW
   2 PRINT TAB 12;"[graphics]"
   3 PRINT TAB 10;"[graphics]"
   4 PRINT TAB 8;"[graphics]"
   5 PRINT TAB 7;"[graphics]"
   6 PRINT TAB 7;"[graphics]"
   7 PRINT TAB 6;"[graphics]"
   8 PRINT TAB 6;"[graphics]"
   9 PRINT TAB 13;"[graphics]"
  10 PRINT TAB 12;"[graphics]"
  11 PRINT TAB 11;"[graphics]"
  12 PRINT TAB 10;"[graphics]"
  13 PRINT TAB 10;"[graphics]"
  14 PRINT TAB 11;"[graphics]"
  15 PRINT TAB 12;"[graphics]"
  16 PRINT "[graphics]"
  20 PRINT
  30 PRINT "[graphics]"
  35 PRINT "[graphics]"
  40 PRINT "[graphics]"
  50 PRINT
  60 PRINT "[graphics]"
  70 PRINT TAB 18;"PREMI NEW-LIN
E"
  80 INPUT U$
  85 CLS
  90 PRINT "CONOSCO CIRCA 60 FUN
GHI TRA I ","PIU"" COMUNI.RISPON
DI ALLE DOMÁN-DE CHE TI FARO"" E
 IO TI DIRO"" ILNOME DEL FUNGO C
HE HAI TROVATO."
  93 FOR T=1 TO 250
  95 NEXT T
 100 CLS
 110 LET K$=""
```

```
 120 PRINT TAB 9;"COLORE CAPPELL
O"
 130 PRINT
 140 PRINT "1) BIANCO O GRIGIO"
 150 PRINT "2) GIALLO O DORATO"
 160 PRINT "3) ROSSO O ARANCIONE
"
 170 PRINT "4) BRUNO O OCRA"
 180 PRINT "5) VIOLETTO O ROSA"
 190 PRINT "6) VIOLA SCURO"
 200 PRINT "7) VERDE OLIVACEO"
 210 PRINT "8) NERO"
 230 GOSUB 1000
 240 PRINT TAB 9;"ORNAMENTO CAPP
ELLO"
 250 PRINT
 260 PRINT "1) LISCIO"
 270 PRINT "2) FIBRILLOSO"
 280 PRINT "3) VERRUCOSO"
 290 PRINT "4) CON PLACCHE"
 300 PRINT "5) VELLUTATO"
 310 PRINT "6) DESQUAMATO"
 320 PRINT "7) SCREPOLATO"
 330 PRINT "8) MARGINE STRIATO"
 340 PRINT "9) [illegible]"
 345 GOSUB 1000
 350 PRINT TAB 9;"COLORE LAMELLE
O PORI"
 355 PRINT
 360 PRINT "1) BIANCO O GRIGIO"
 370 PRINT "2) GIALLO"
 380 PRINT "3) ROSSO SCURO"
 390 PRINT "4) OCRA"
 400 PRINT "5) ROSA O VIOLETTO"
 410 PRINT "6) VIOLA SCURO"
 420 PRINT "7) VERDE GIALLASTRO"
 430 PRINT "8) NERO"
 435 GOSUB 1000
 440 PRINT TAB 9;"FORMA IMENIO"
 450 PRINT
 460 PRINT "1) LAMELLE LIBERE"
 470 PRINT "2) LAMELLE ANNESSE"
 480 PRINT "3) LAMELLE ADNATE"
```

```
 490 PRINT "4) LAMELLE DECORRENT
I"
 500 PRINT "5) LAMELLE SINUATE"
 510 PRINT "6) VENATURE"
 520 PRINT "7) TUBULI"
 530 GOSUB 1000
 540 PRINT TAB 9;"COLORE GAMBO"
 550 PRINT
 560 PRINT "1) BIANCO"
 570 PRINT "2) GIALLO"
 580 PRINT "3) ARANCIO"
 590 PRINT "4) BRUNO O ROSSO"
 600 PRINT "5) VIOLETTO"
 610 PRINT "6) VIOLA SCURO"
 620 PRINT "7) VERDASTRO"
 630 PRINT "8) NERO"
 640 GOSUB 1000
 650 PRINT TAB 9;"ORNAMENTO GAMB
O"
 660 PRINT
 670 PRINT "1) RETICOLATO"
 680 PRINT "2) PUNTEGGIATO"
 690 PRINT "3) ZONATO"
 700 PRINT "4) VOLVA A SACCO E A
NELLO"
 710 PRINT "5) VOLVA NON A SACCO
 E ANNELLO        LISCIO"
 720 PRINT "6) VOLVA NON A SACCO
 E ANELLO         STRIATO"
 730 PRINT "7) SOLO L""ANELLO"
 740 PRINT "8) LISCIO O FIBRILLO
SO"
 750 GOSUB 1000
 760 PRINT TAB 9;"COLORE CARNE"
 770 PRINT
 780 PRINT "1) BIANCO O GRIGIO"
 790 PRINT "2) INGIALLISCE"
 800 PRINT "3) ARROSSA"
 810 PRINT "4) GIALLO"
 820 PRINT "5) GIALLO IMBLUENTE"
 830 PRINT "6) LATTE ROSSO"
 840 PRINT "7) LATTE BIANCO"
 850 PRINT "8) SCURO"
 860 GOSUB 1000
 870 PRINT TAB 9;"SAPORE"
 880 PRINT
 890 PRINT "1) DOLCE FUNGINO"
 900 PRINT "2) FARINA FRESCA"
 910 PRINT "3) FARINA RANCIDA"
 920 PRINT "4) AMARO"
 930 PRINT "5) ACRE"
 940 PRINT "6) DI PESCE"
 950 GOSUB 1000
 960 GOTO 1500
1000 PRINT "0) INDEFINIBILE"
1005 IF INKEY$="" THEN GOTO 1005
1010 LET U$=INKEY$
1012 IF U$="" THEN GOTO 1010
1015 LET K$=K$+U$
1020 CLS
1030 RETURN
1500 PRINT "TRA TUTTE LE CARATTE
RISTICHE DA ";"TE ELENCATE INDIC
ANE DUE DI CUI ";"SEI ASSOLUTAME
TE CERTO:"
1510 PRINT
1520 PRINT "1) COLORE CAPPELLO"
1530 PRINT "2) ORNAGENTO CAPPELL
O"
1540 PRINT "3) COLORE LAMELLE O
TUBULI"
1550 PRINT "4) FORMA IMENIO"
1560 PRINT "5) COLORE GAMBO"
1570 PRINT "6) ORNAMENTO GAMBO"
1580 PRINT "7) COLORE CARNE"
1590 PRINT "8) SAPORE"
1595 PRINT AT 15,7;"=>";AT 15,17
;"=>"
1600 IF INKEY$="" THEN GOTO 1600
1610 LET M=VAL INKEY$
1620 IF M<1 OR M>8 THEN GOTO 160
0
1625 PRINT AT 15,10;M
1630 IF INKEY$="" THEN GOTO 1630
1640 LET N=VAL INKEY$
1650 IF N<1 OR N>8 THEN GOTO 163
0
1660 IF N=M THEN GOTO 1630
1670 PRINT AT 15,20;N
1680 FOR T=1 TO 50
1690 NEXT T
2000 LET R=0
2003 CLS
2005 FAST
2010 LET K=0
2020 FOR A=1 TO 60
2025 IF N$(A)(M TO M)<>K$(M TO M
) OR N$(A)(N TO N)<>K$(N TO N) T
HEN GOTO 2110
2030 LET P=0
2040 FOR B=1 TO 8
2050 IF N$(A)(B TO B)=K$(B TO B)
 THEN LET P=P+1
2060 NEXT B
2080 IF P>K THEN LET J=A
2090 IF P>K THEN LET R=1
2100 IF P=K THEN LET R=R+1
2105 IF P>K THEN LET K=P
2110 NEXT A
2112 SLOW
2113 IF K<4 THEN GOTO 3000
2120 IF R>1 THEN PRINT "DATI INS
UFFICIENTI O ERRATI.","CI SONO "
;R;" FUNGHI CHE SI AVVICI- ";"NA
NO ALLA TUA DESCIZIONE."
2125 IF R>1 THEN GOTO 5000
2130 IF R=1 THEN PRINT "IL FUNGO
 CHE PIU"" SI AVVICINA ","(";12.
5*K;"%)";" ALLA TUA DESCRIZION
E","SI CHIAMA:";AT 5,5;F$(J)
2135 IF J=1 OR J=2 OR J=59 THEN
GOTO 2500
2140 GOTO 5000
2500 PRINT
2510 PRINT
2520 PRINT TAB 11;CHR$ 134;CHR$
0;CHR$ 0;CHR$ 0;CHR$ 0;CHR$ 6
2530 PRINT TAB 11;CHR$ 0;CHR$ 7;
CHR$ 128;CHR$ 128;CHR$ 132
2540 PRINT TAB 11;CHR$ 0;CHR$ 12
8;CHR$ 128;CHR$ 128;CHR$ 128
2550 PRINT TAB 11;CHR$ 5;CHR$ 0;
CHR$ 130;CHR$ 129;CHR$ 0;CHR$ 13
4
2560 GOTO 5000
2999 STOP
3000 PRINT "MI DISPIACE.NON CONO
SCO QUESTO  FUNGO."
5000 PRINT AT 21,8;"ANCORA? S/N"
5010 IF INKEY$="" THEN GOTO 5010
5030 IF INKEY$="S" THEN GOTO 100
5035 CLS
5040 PRINT AT 10,8;"ARRIVEDERCI"
6000 STOP
9997 STOP
9998 SAVE "FUNGHI"
9999 GOTO 1
```

Seguito figura 3.

HP-Stat

Un programma per l'adattamento di alcune variabili statistiche a valori campionari

di **G. Taroni**

Il programma consente la stima dei parametri di scala e di locazione per alcune variabili statistiche comunemente impiegate per la descrizione di fenomeni idrologici e nella teoria della affidabilità; alcuni controlli di adattamento (goodness of fit) (empirici) della variabile statistica prescelta per l'interpretazione del fenomeno quantitativo in studio.

Le variabili trattate dal programma sono le seguenti:
Gumbel per la distribuzione dei massimi,
Logistica,
Esponenziale,
Normale,
Lognormale,
Cauchy,
Gumbel per la distribuzione dei minimi,
Weibull,
Pareto,
Frechét.
Tutte caratteristiche da due parametri, di locazione e di scala.
Il calcolo dei parametri si è svolto con alcune varianti del metodo "best line fitting" per poter essere applicato anche a campioni suddivisi in classi di frequenze.
Il programma è implementato su Hewlett Packard 85; non ricorre alla Advanced Programming ROM anche se, per maggiore chiarezza, sono riportate come istruzioni commento (!) alcune CALL in quanto ne facilitano la trascrizione in altri linguaggi.

Alcune note

Se la distribuzione scelta è la Lognormale, la Weibull, la Pareto o la Frechét, il programma pone in X () i logaritmi dei valori campionari.
Per condurre le verifiche che comportano il calcolo dei logaritmi è bene accertare che tutti gli X () siano maggiori di 0. In questo caso, per il calcolo dei quantili associati ai livelli di "probabilità" richiesti, rispondere con S all'opzione logaritmica.

In generale, per ottenere le quantità X teoriche (calcolate cioè supponendo che X sia realmente tratto dalla variabile statistica scelta) associate ai livelli di probabilità che ci interessano, basta impostare un valore tra 0 e 1, estremi esclusi. Se si è interessati ai quantili associati ad un periodo di ritorno Q, basta impostare il numero (Q-1)/Q. (19/20 per 20 unità temporali).
Se la numerosità campionaria non è troppo modesta si può giudicare l'accostamento della variabile statistica scelta al campione X tramite il test Chiquadrato se i valori sono suddivisi in classi o tramite il grafico altrimenti.

Nel primo caso l'accostamento della variabile statistica scelta è giudicato in funzione del valore fornito dal test Chiquadrato che va confrontato con il valore critico della omonima variabile con K1-1 gradi di libertà.
Nel grafico invece sono riportati gli intervalli di confidenza al 5% del test di Kolmogoroff relativi alla stima della funzione di ripartizione in corrispondenza ai K0 valori della variabile X( ). Tali intervalli sono poi uniti con tratteggi continui e si ottengono così linee di massimo e di minimo.

È inoltre riportata la funzione di ripartizione L( ). Per accettare l'ipotesi di adeguamento della variabile scelta, la funzione di ripartizione L( ) deve intersecare tutti gli intervalli della figura. Se ciò non accade non è probabile (5%) che il campione X( ) sia effettivamente tratto dalla variabile statistica ipotizzata. Il grafico successivo riporta i valori campionari X( ), F( ) in scala di probabilità della variabile statistica scelta, tratteggiata sull'asse Y.

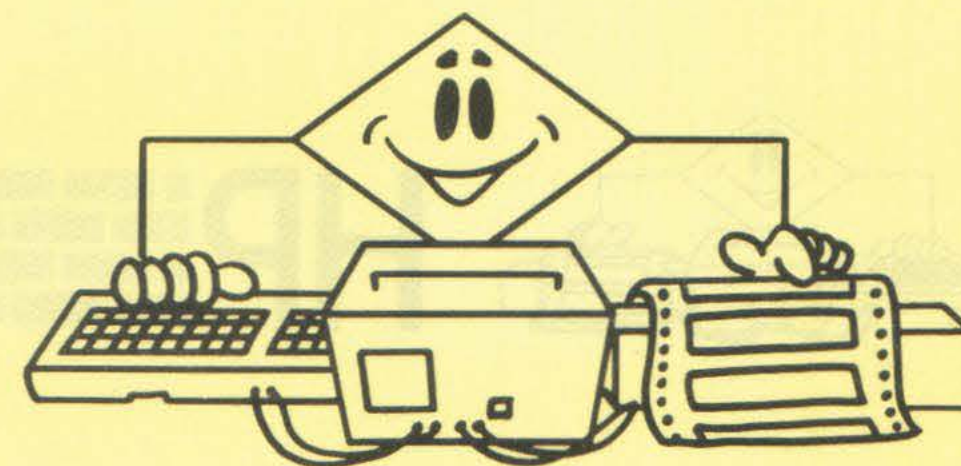
Per un accostamento "perfetto" i punti devono disporsi sulla retta tratteggiata. Per un più approfondito esame statistico dei problemi delineati, si rimanda alla bibliografia.

Bibliografia

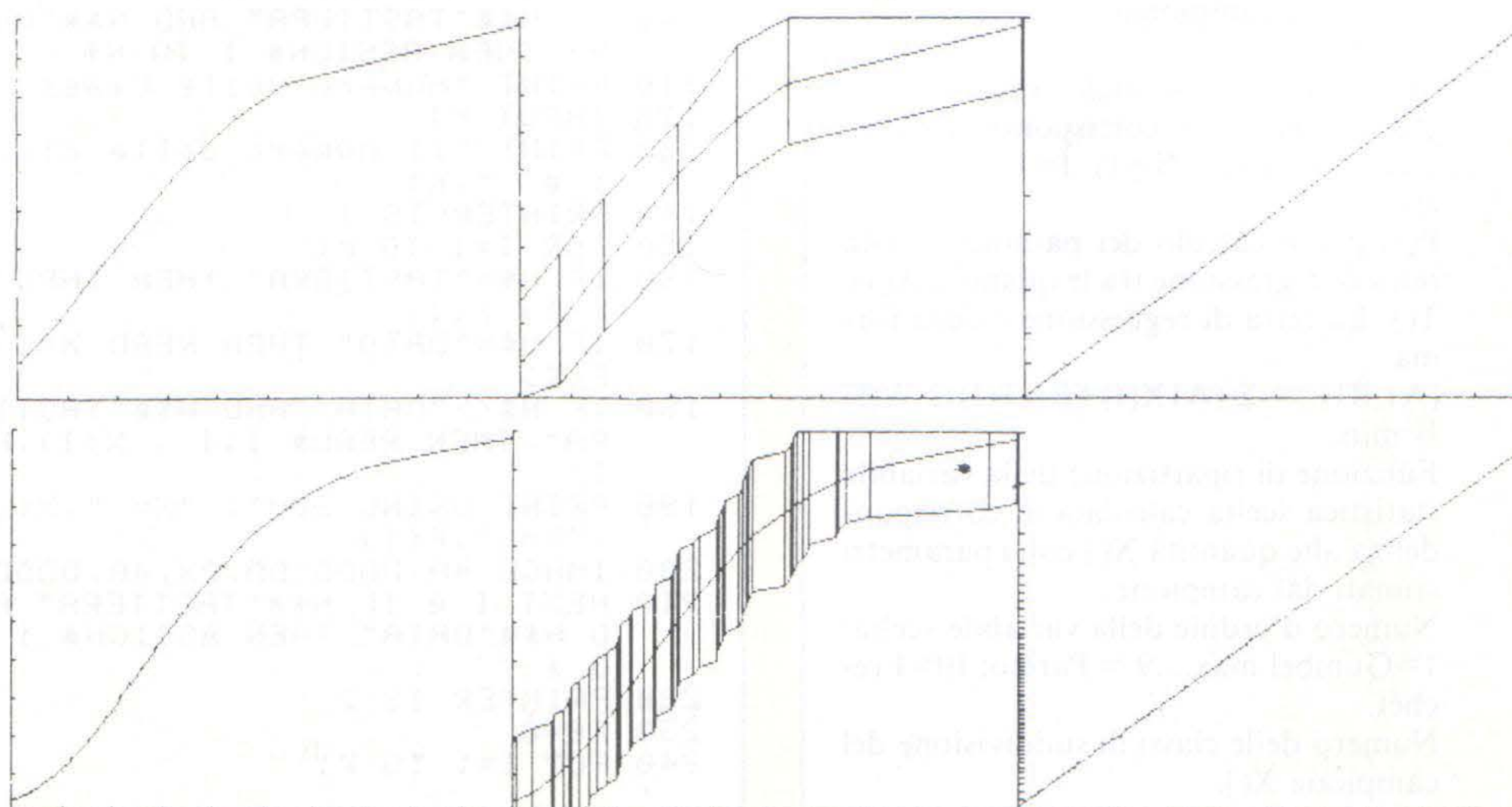
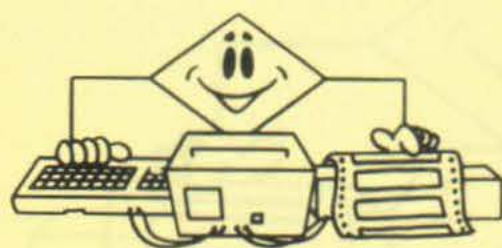
(1) Azzalini A., Pesarin F. (1976): Verifica statistica di ipotesi funzionale non parametrica. L'Elaborazione Automatica 2,3.
(2) L. Carbognin, G. Taroni (1982). Verifica statistica di ipotesi funzionali con parametri stimati. Applicazione agli eventi delle acque alte di Venezia. In corso di pubblicazione su Statistica Applicata. Vol. 3 - 1983.
(3) Pesarin F. (1981): Un test per il controllo statistico di curve di affidabilità dipendenti da due parametri, con dati censurati. Atti del Convegno della S.I.S., 271, 84.

LE SUBROUTINE

FNA	Calcola il vettore W() relativo ai pesi da attribuire all'interpolazione per la stima dei parametri per campioni censurati come proposto in (3).
FNP	Calcola i parametri A1 e B1 della retta di regressione con il metodo dei minimi quadrati.
FNR	Calcola il coefficiente di correlazione lineare.
FNT	Calcola i valori "tipo" o quantili tratti dalla variabile statistica scelta.
NDTRI	Per i quantili della variabile normale.
FNF	Calcola la funzione di ripartizione della variabile statistica scelta.
NDTR	Per la ripartizione della normale.
FNH$	Funzione della variabile statistica scelta.
GRAF	Esegue il grafico con scale opportune: asse Y tra 0 e 1, asse X tra X(1), X(KO).
QUANT	Calcola i quantili e i periodi di ritorno associati ai livelli di probabilità introdotti da tastiera.

HP

```
NOME DEL FILE ?
TASTIERA DATA FILE ?
?
TASTIERA
TASTIERA
numero delle classi
?
7
il numero delle classi e'  7
?
34.3,5
X=    34.30  F=     5
?
39.7,7
X=    39.70  F=     7
?
46,8
X=    46.00  F=     8
?
56,11
X=    56.00  F=    11
?
64.1,11
X=    64.10  F=    11
?
71.2,5
X=    71.20  F=     5
?
103,8
X=   103.00  F=     8
NUMEROSITA' CAMPIONARIA  55
TIPO DI DISTRIBUZIONE
GUMBEL MASSIMI           1
LOGISTICA                2
ESPONENZIALE             3
NORMALE                  4
LOGNORMALE               5
CAUCHY                   6
GUMBEL MINIMI            7
WEIBULL                  8
PARETO                   9
FRECHE'T                10
?
1
GUMBEL MASSIMI
 VALORI CENSURATI
?
NO
NO
X=    34.300 VC TIPO    -.882
X=    39.700 VC TIPO    -.432
X=    46.000 VC TIPO    -.029
X=    56.000 VC TIPO     .525
X=    64.100 VC TIPO    1.246
X=    71.200 VC TIPO    1.742
X=   103.000 VC TIPO    4.016
```

```
PESI Wi=Fi:S/N
?
N
N
A1=        .069  B1=     -3.223
Var.res **  010119415134  **
CORR.LIN=          997444
Valori della f.di ripartizione
Pr(X<=   34.30)      .0951
Pr(X<=   39.70)      .1977
Pr(X<=   46.00)      .3502
Pr(X<=   56.00)      .5908
Pr(X<=   64.10)      .7402
Pr(X<=   71.20)      .8317
Pr(X<=  103.00)      .9797
  *** FREQUENZE ***
OSSERVATE   TEORICHE
    5          5.229
    7          5.645
    8          8.385
   11         13.237
   11          8.214
    5          5.033
    8          9.257
CIQUAD.= 1.8470890719
LOCAZ=    46.70  SCALA=    14.49
CALCOLO DEI QUANTILI: .5 PER LA
MEDIANA
VALORI 0<X<1:<=0PER TORNARE NEL
MAIN
OPZIONE LOGARITMICA?(S/N)
?
N
   QUANTILE    PROB    P.RITORNO
?
.5
   52.01       .500        2.00
?
.9
   79.30       .900       10.00
?
.95
   89.73       .950       20.00
?
.98
  103.23       .980       50.00
?
54/55
  104.62       .982       55.00
?
.99
  113.34       .990      100.00
?
-1
G PER IL GRAFICO
?
G
```

Figura 1 - Due esempi di stampa delle opzioni possibili. La richiesta dei grafici produce tre uscite. La prima è la ripartizione della var. scelta L(). Tratto continuo. Ripartizione empirica frequenze relative cumulate M()/(N+1). Tratto a puntini.
La seconda riporta gli intervalli di stima della funzione di ripartizione campionaria in corrispondenza degli X(). Se la curva L() interseca tutti gli intervalli verticali è superato il test grafico.
La terza è una carta di probabilità della variabile scelta: quantili della variabile tipo in ordinata; variabile campionaria in ascissa.

HP

Seguito figura 1.

```
NOME DEL FILE ?
ONDE
numero delle classi
il numero delle classi e'  55
NUMEROSITA' CAMPIONARIA  55
TIPO DI DISTRIBUZIONE

GUMBEL MASSIMI              1
LOGISTICA                   2
ESPONENZIALE                3
NORMALE                     4
LOGNORMALE                  5
CAUCHY                      6
GUMBEL MINIMI               7
WEIBULL                     8
PARETO                      9
FRECHE'T                    10

GUMBEL MASSIMI

 VALORI CENSURATI
NO
PESI Wi=Fi:S/N
N
A1=      .068  B1=   -3.205
Var.res **  .185561085218  **
CORR.LIN=        .993732
LOCAZ=   47.30  SCALA=   14.75
CALCOLO DEI QUANTILI: .5 PER LA
MEDIANA
VALORI 0<X<1:<=0PER TORNARE NEL
MAIN
OPZIONE LOGARITMICA?(S/N)
   QUANTILE    PROB    P.RITORNO
   52.70       .500       2.00
   80.50       .900      10.00
   91.12       .950      20.00
  104.87       .980      50.00
  106.29       .982      55.00
  115.17       .990     100.00
  125.43       .995     200.00
G PER IL GRAFICO
G
```

LE QUANTITA' DEL PROGRAMMA

N — Numero di elementi campionari.

X() — Vettore campionario: X(1) <= X(k): <= N; i valori X() sono gli estremi superiori della suddivisione in classi. X(1) <= <= X(k).

F() — Frequenze campionarie associate ai valori di X():X(1) con freq. F(1) .... Se il campione non è suddiviso in classi, ogni F() è pari a 1.

M() — Frequenze campionarie cumulate: M(J) = Σ F(I); I=1, ...,J.

T() — Quantità della variabile statistica prescelta calcolata in corrispondenza delle quantità: M(I)/(N+1) I=1,.. numero classi.

W() — Pesi per il calcolo dei parametri della retta di regressione tra le quantità X() e T(). La retta di regressione è della forma:
(A1,B1): = Σ (A1X(I)+B1-T(I)↑2 W(I) = min.

L() — Funzione di ripartizione della variabile statistica scelta calcolata in corrispondenza alle quantità X() con i parametri stimati dal campione.

L — Numero d'ordine della variabile scelta: 1=Gumbel max....9 = Pareto; 10=Frechét.

K1 — Numero delle classi di suddivisione del campione X().

N$ — Nome del FILE. Viene posto da input. Se i dati sono su nastro si imposta il nome del file. Il FILE deve essere suddiviso in K1 record.
Il record a sua volta contiene i valori X(I), F(I). (16 byte per record).
Se i dati si danno da input si risponde TASTIERA o DATA se sono contenuti nell'omonima istruzione.

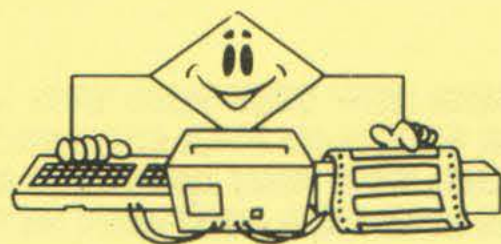
A1$ — Viene posto da input e assume le modalità SI, NO. SI se i valori sono censurati a destra; si conosce cioè solo il numero degli X(), F(), superiori a X(K1). NO altrimenti.

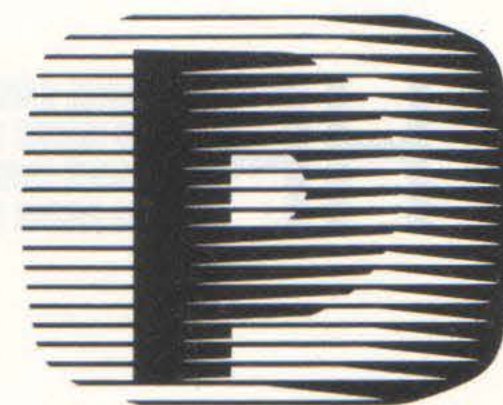
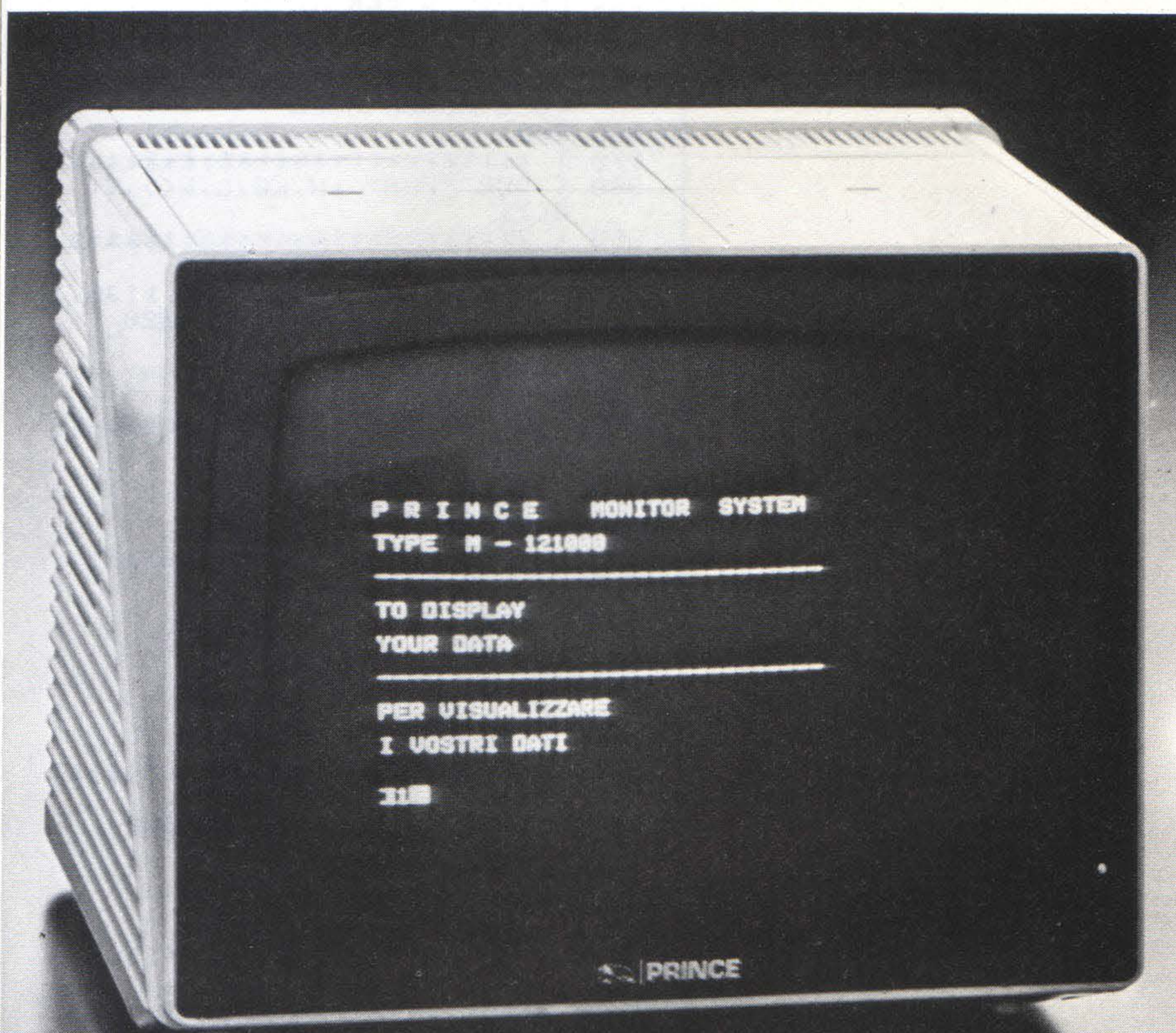
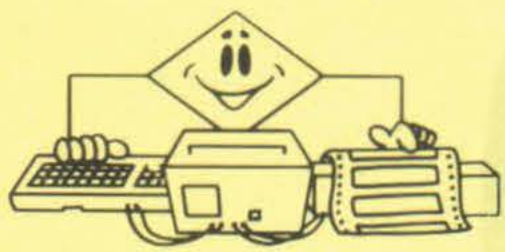
KO — Viene posto dal programma =K1-1 se A1$=SI, pari a K1 altrimenti.

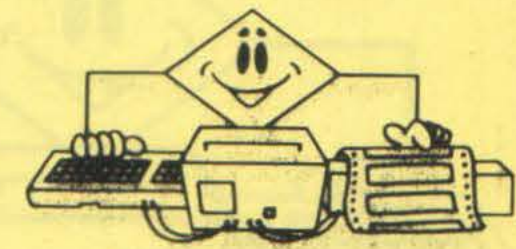
P2$ — Assume le modalità SI, NO. Con SI l'interpolazione tra le X() e T() viene svolta con pesi W() proporzionali alle frequenze F(). Con NO sono calcolati come proposto in (3) per campioni censurati.
Si consiglia quest'ultimo metodo se il campione è censurato.

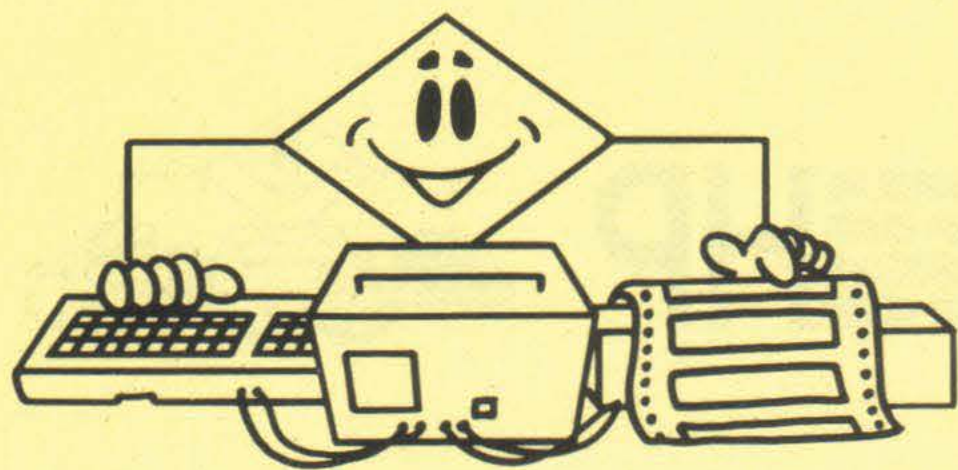
P$ — Viene posto da input. Se G viene condotto un test di adattamento grafico utilizzando il criterio Kolmogoroff.

Figura 2 - Il listato BASIC.

```
 10 OPTION BASE 1
 20 SHORT X(90),T(90),W(90),L(90
    )
 30 INTEGER N,K1,K0,L,L0
 40 INTEGER F(90),M(90)
 50 DIM A1$[2],N$[16],P$[2],P2$[
    2],L$[1]
 60 !\adattamento di una variabi
    le statistica a valori campi
    onari
 70 V0=1.36 ! VALORE AL 5% DEL T
    EST DI KOLMOGOROFF; 1.63PER
    1%
 80 PRINTER IS 2
 90 PRINT "NOME DEL FILE ?" @ DI
    SP "TASTIERA DATA FILE ?" @
    INPUT N$@ PRINT N$
100 IF N$#"TASTIERA" AND N$#"DAT
    A" THEN ASSIGN# 1 TO N$
110 PRINT "numero delle classi"
120 INPUT K1
130 PRINT "il numero delle class
    i e' ";K1
140 PRINTER IS 1
150 FOR I=1 TO K1
160 IF N$="TASTIERA" THEN INPUT
    X(1),F(I)
170 IF N$="DATA" THEN READ X(I),
    F(I)
180 IF N$<>"DATA" AND N$#"TASTIE
    RA" THEN READ# 1,I ; X(I),F(
    I)
190 PRINT USING 200 ; "X= ",X(I)
    ,"F= ",F(I)
200 IMAGE 4A,DDDD.DD,2X,4A,DDDD
210 NEXT I @ IF N$#"TASTIERA" AN
    D N$#"DATA" THEN ASSIGN# 1 T
    O *
220 PRINTER IS-2
230 F0=0
240 FOR I=1 TO K1
```

Seguito figura 2.

```
250 F0=F0+F(I)
260 M(I)=INT(F0)
270 NEXT I
280 N=M(K1)
290 PRINT "NUMEROSITA' CAMPIONAR
    IA ";N
300 PRINT "TIPO DI DISTRIBUZIONE
    "
310 PRINT " "
320 FOR I=1 TO 10
330 PRINT FNH$(I),I
340 NEXT I
350 INPUT L
360 PRINT " "
370 PRINT FNH$(L)
380 PRINT " "
390 PRINT " VALORI CENSURATI"
400 INPUT A1$
410 PRINT A1$
420 IF A1$="NO" THEN K0=K1 ELSE
    K0=K1-1
430 FOR I=1 TO K0
440 IF L=5 OR L=9 OR L=8 OR L=10
     THEN X(I)=LOG(X(I)) ELSE I=
    K0
450 NEXT I @ L0=(L=5)*4+(L=8)*7+
    (L=9)*3+(L=10)*8
460 IF L0>0 THEN L=L0
470 PRINTER IS 1
480 FOR I=1 TO K0 @ W(I)=0
490 X1=M(I)/(N+1)
500 Z=X1
510 GOSUB 1760 @ T(I)=F0 ! CALL
    "FNT"(X1,F0,L)
520 PRINT USING 530 ; X(I),T(I)
530 IMAGE "X=",1X,DDDD.DDD,1X,"V
    C TIPO",1X,DDD.DDD
540 NEXT I
550 PRINT " " @ PRINTER IS 2
560 PRINT "PESI Wi=Fi:S/N" @ INP
    UT P2$@ PRINT P2$
570 IF P2$[1,1]#"S" THEN GOSUB 9
    80 ELSE GOSUB 1370 ! CALL "F
    NA" ( N,K0,L,W(),T(),M() )
580 !
590 GOSUB 1390 ! CALL "FNP" ( N,
    K0,X(),W(),T(),A1,B1,P5,P6)
600 !
610 PRINT USING 620 ; "A1=",A1,"
    B1=",B1
620 IMAGE 3A,2X,DDD.DDD,2X,3A,2X
    ,DDD.DDD
630 A0=0 @ FOR I=1 TO K0 @ A0=A0
    +(A1*X(I)+B1-T(I))^2*W(I) @
    NEXT I @ A0=A0*(N+1)/(M(K0)-
    2)
640 PRINT "Var.res ";"** ";A0;"
    **"
650 GOSUB 1540 ! CALL "FNR" ( N,
    K0,F0,X(),W(),T() )
660 PRINT USING 680 ; "CORR.LIN=
    ",ABS(F0) @ PRINTER IS 1
670 PRINT @ PRINT "______________
    __"
680 IMAGE 10A,2X,DDD.DDDDDD
690 PRINT @ PRINT "Valori della
    f.di ripartizione" @ PRINT
700 FOR I=1 TO K0 @ W0=X(I)
710 W=W0
720 GOSUB 2100 ! CALL "FNF" ( W0
    ,A1,B1,L,Z1 )
730 Z1=F0
740 L(I)=Z1
750 PRINT USING 760 ; X(I),Z1
760 IMAGE "Pr(X<=",1X,DDDD.DD,")
    ",2X,DD.DDDD
770 NEXT I
780 PRINT
790 !
```

Seguito figura 2.

```
800 PRINT "  *** FREQUENZE *** "
810 PRINT "OSSERVATE  TEORICHE"
820 X1=0 @ X2=FNL(1)
830 PRINT USING 900 ; F(1),X2 @
    IF X2>0 AND K1<N THEN X1=X1+
    (F(1)-X2)^2/X2
840 FOR I=2 TO K1 @ X2=FNL(I)-FN
    L(I-1) @ PRINT USING 900 ; F
    (I),X2
850 IF X2>0 AND K1<N THEN X1=X1+
    (F(I)-X2)^2/X2
860 NEXT I @ PRINTER IS 2 @ IF K
    1<N THEN PRINT "CIQUAD.=";X1
870 X1=-(B1/A1) @ X2=1/A1 @ PRIN
    T USING 880 ; X1,X2
880 IMAGE "LOCAZ=",1X,DDDD.DD,2X
    ,"SCALA=",1X,DDDD.DD
890 GOSUB 2900
900 IMAGE DDDDD,5X,DDDD.DDD
910 PRINTER IS 2
920 PRINT "G PER IL GRAFICO "
930 INPUT P$
940 PRINT P$
950 IF P$[1,1]="G" THEN GOSUB 26
    40 @ GOSUB 3020
960 END
970 ! *************************
980 ! SUB "FNA" (N,K0,L,W(),T(),
    M())
990 ! *************************
1000 FOR I=1 TO K0
1010 ON L GOSUB 1050,1090,1130,1
     170,1210,1230,1280,1320
1020 W(I)=R9
1030 NEXT I
1040 RETURN
1050 W1=(N+2)*M(I)*(LOG(M(I))-LO
     G(N+1))^2
1060 W2=(N+1-M(I))*(N+1)
1070 R9=W1/W2
1080 RETURN
1090 W1=(N+2)*M(I)*(N+1-M(I))
1100 W2=(N+1)^3
1110 R9=W1/W2
1120 RETURN
1130 W1=(N+2)*(N+1-M(I))
1140 W2=(N+1)*M(I)
1150 R9=W1/W2
1160 RETURN
1170 W1=(N+1)*(N+2)*EXP(-(T(I)*T
     (I)))
1180 W2=M(I)*(N+1-M(I))*PI*2
1190 R9=W1/W2
1200 RETURN
1210 GOSUB 1170
1220 RETURN
1230 W0=(N+1)*(N+2)*COS(M(I)*PI/
     (N+1)-PI/2)^4
1240 W1=M(I)*(N+1-M(I))
1250 R9=W0/W1
1260 R9=R9/PI^2
1270 RETURN
1280 W1=(N+2)*(N+1-M(I))*(LOG(N+
     1)-LOG(N+1-M(I)))^2
1290 W2=M(I)*(N+1)
1300 R9=W1/W2
1310 RETURN
1320 W1=(N+2)*M(I)*(LOG(M(I))-LO
     G(N+1))^2
1330 W2=(N+1-M(I))*(N+1)
1340 R9=W1/W2
1350 RETURN
1360 ! *************************
     *
1370 FOR I=1 TO K0 @ W(I)=F(I)/N
      @ NEXT I @ RETURN
1380 ! *************************
     **
1390 ! SUB "FNP" (N,K0,Y(),W(),T
     (),A1,B1,P5,P6)
1400 ! *************************
     *
1410 P1,P2,P3,P4,P5,P6=0
1420 FOR I=1 TO K0
1430 P1=P1+X(I)*X(I)*W(I)
1440 P2=P2+X(I)*W(I)
1450 P3=P3+T(I)*X(I)*W(I)
1460 P5=P5+W(I) @ P6=P6+W(I)*T(I
     )
1470 NEXT I
1480 P4=P2
1490 D=P1*P5-P4*P2
1500 A1=(P3*P5-P6*P2)/D
1510 B1=(P1*P6-P4*P3)/D
1520 RETURN
1530 ! *************************
     *
1540 ! SUB "FNR" (N,K0,F0,X(),W(
     ),T())
1550 ! *************************
     *
1560 X1,X2,V,Y1,Y2,R,W0=0
1570 FOR I=1 TO K0
1580 X1=X1+X(I)*W(I)
1590 X2=X2+X(I)*X(I)*W(I)
1600 V=X(I)*T(I)*W(I)+V
1610 Y1=Y1+T(I)*W(I)
1620 Y2=Y2+T(I)*T(I)*W(I)
1630 W0=W0+W(I)
1640 NEXT I
1650 X1=X1/W0
1660 X2=X2/W0
1670 Y1=Y1/W0
1680 Y2=Y2/W0
1690 V=V/W0
1700 X2=X2-X1*X1
1710 Y2=Y2-Y1*Y1
1720 R=(V-X1*Y1)/SQR(Y2*X2)
1730 F0=R
1740 RETURN
1750 ! *************************
     *
1760 ! SUB "FNT" (Z,F0,L)
1770 ! *************************
     *
1780 ON L GOTO 1790,1810,1830,18
     50,1850,1900,1920,1940
1790 F0=-LOG(-LOG(Z))
1800 RETURN
1810 F0=-LOG((1-Z)/Z)
1820 RETURN
1830 F0=-LOG(1-Z)
1840 RETURN
1850 P=Z @ M=0 @ S=1
1860 GOSUB 1970 ! CALL "NDTRI" (
     Z,0,1,Z3 )
1870 Z3=X
1880 F0=Z3
1890 RETURN
1900 F0=TAN(PI*Z-PI/2)
1910 RETURN
1920 F0=LOG(-LOG(1-Z))
1930 RETURN
1940 F0=LOG(-LOG(Z))
1950 RETURN
1960 ! *************************
     *
1970 ! SUB "NDTRI" (P,M,S,X)
1980 ! *************************
     *
1990 IF P<0 THEN PRINT "PRNEG" @
      RETURN
2000 IF P=0 THEN X=-(100*S)+M @
     RETURN
2010 IF P>1 THEN PRINT "PRMAG1"
     @ RETURN
2020 IF P=1 THEN X=100*S+M @ RET
     URN
2030 D=P
2040 IF D>.5 THEN D=1-D
2050 T2=LOG(1/(D*D)) @ T=SQR(T2)
2060 X=T-(2.515517+.802853*T+.01
     0328*T2)/(1+1.432788*T+.189
     269*T2+.001308*T*T2)
```

```
2070 IF P<=.5 THEN X=-(X*S)+M EL
     SE X=X*S+M
2080 RETURN
2090 ! *************************
     *
2100 ! SUB "FNF" (W,A1,B1,L,F0)
2110 ! *************************
     *
2120 ON L GOTO 2130,2160,2190,22
     50,2250,2320,2340,2360
2130 W1=A1*W+B1
2140 F0=EXP(-EXP(-W1))
2150 RETURN
2160 W1=A1*W+B1
2170 F0=1/(1+EXP(-W1))
2180 RETURN
2190 W1=A1*W+B1
2200 IF W1<0 THEN 2230
2210 F0=1-EXP(-W1)
2220 RETURN
2230 F0=0
2240 RETURN
2250 S=1/A1
2260 M=-(S*B1) @ X=W
2270 GOSUB 2390 ! CALL "NDTR" (
     W,M,S,W1 )
2280 W1=P
2290 F0=W1
2300 !
2310 RETURN
2320 F0=(ATN(A1*W+B1)+PI/2)/PI
2330 RETURN
2340 F0=1-EXP(-EXP(A1*W+B1))
2350 RETURN
2360 F0=EXP(-EXP(A1*W+B1))
2370 RETURN
2380 ! *************************
     *
2390 ! SUB "NDTR" (X,M,S,P)
2400 ! *************************
     *
2410 A=ABS((X-M)/S)
2420 T=1/(1+.2316419*A)
2430 D=.3989423*EXP(-((X-M)*(X-M
     )/(S*S*2)))
2440 P=1-D*T*((((1.330274*T-1.82
     1256)*T+1.781478)*T-.356563
     8)*T+.3193815)
2450 IF X<M THEN P=1-P
2460 RETURN
2470 ! *************************
     *
2480 DEF FNH$(I)
2490 ! *************************
     *
2500 IF I=1 THEN FNH$="GUMBEL MA
     SSIMI"
2510 IF I=2 THEN FNH$="LOGISTICA
     "
2520 IF I=3 THEN FNH$="ESPONENZI
     ALE"
2530 IF I=4 THEN FNH$="NORMALE"
2540 IF I=5 THEN FNH$="LOGNORMAL
     E"
2550 IF I=6 THEN FNH$="CAUCHY"
2560 IF I=7 THEN FNH$="GUMBEL MI
     NIMI"
2570 IF I=8 THEN FNH$="WEIBULL"
2580 IF I=9 THEN FNH$="PARETO"
2590 IF I=10 THEN FNH$="FRECHE'T
     "
2600 IF I<1 OR I>10 THEN FNH$="E
     RRORE INPUT"
2610 FN END
2620 ! *************************
     *
2630 ! SUB GRAF(X(),K0,M(),L())
2640 REM ..SUB GRAFICI
2650 ! *************************
     *
2660 GCLEAR @ SCALE X(1),X(K0),0
     ,1
2670 XAXIS 0,(X(K0)-X(1))/10 @ Y
     AXIS X(1),.1
2680 FOR I=1 TO K0
2690 MOVE X(I),M(I)/(N+1) @ PLOT
     X(I),M(I)/(N+1)
2700 NEXT I
2710 MOVE X(1),L(1) @ FOR I=1 TO
     K0 @ DRAW X(I),L(I) @ NEXT
     I @ COPY
2720 FOR I=1 TO K0
2730 MOVE X(I),MAX(0,-(V0/SQR(N)
     )+M(I)/N)
2740 DRAW X(I),MIN(1,V0/SQR(N)+(
     M(I)-1)/N)
2750 PENUP
2760 NEXT I
2770 !
2780 MOVE X(1),MIN(1,(M(1)-1)/N+
     V0/SQR(N))
2790 FOR I=1 TO K0 @ DRAW X(I),M
     IN(1,(M(I)-1)/N+V0/SQR(N))
2800 NEXT I @ MOVE X(1),MAX(0,M(
     1)/N-V0/SQR(N))
2810 FOR I=1 TO K0 @ DRAW X(I),M
     AX(0,M(I)/N-V0/SQR(N))
2820 NEXT I @ COPY
2830 RETURN
2840 ! *************************
     *
2850 DEF FNL(J)
2860 IF J<=K0 THEN FNL=N*L(J)
2870 IF J=K1 THEN FNL=N
2880 FN END
2890 ! *************************
2900 ! SUB QANT: QUANTILI E PERI
     ODI DI RITORNO
2910 PRINT "CALCOLO DEI QUANTILI
     : .5 PER LA MEDIANA"
2920 PRINT "VALORI 0<X<1:<=0PER
     TORNARE NEL MAIN"
2930 PRINT "OPZIONE LOGARITMICA?
     (S/N)"
2940 INPUT L$[1,1]
2950 PRINT "   QUANTILE   PROB
     P.RITORNO"
2960 INPUT Z@ IF Z<=0 OR Z>=1 TH
     EN 3010
2970 GOSUB 1770 @ V=X2*F0+X1 @ I
     F L$[1,1]="S" THEN V=EXP(V)
2980 PRINT USING 2990 ; V,Z,1/(1
     -Z)
2990 IMAGE 1X,DDDD.DD,4X,DD.DDD,
     3X,DDDDD.DD
3000 GOTO 2960
3010 RETURN ! -----------------
     --
3020 ! GRAF2: X() T()
3030 GCLEAR @ X8=MIN(T(1),T(K0))
     @ X9=MAX(T(1),T(K0))
3040 SCALE X(1),X(K0),X8,X9
3050 XAXIS X8,(X(K0)-X(1))/10
3060 YAXIS X(1)
3070 FOR I=1 TO K0 @ MOVE X(I),T
     (I) @ PLOT X(I),T(I) @ NEXT
     I
3080 MOVE X(1),T(1)
3090 FOR I=1 TO K0 @ DRAW X(I),A
     1*X(I)+B1 @ NEXT I
3100 FOR I=1 TO K0 @ MOVE X(1),T
     (I)
3110 PLOT X(1),T(I) @ IDRAW (X(K
     0)-X(1))/100,0
3120 NEXT I
3130 RETURN
3140 DATA 28.9,2,34.3,3,37.1,4,3
     9.7,3,42.2,2,44.4,4,46,2,49
     .6,4,53.1,5,56,2,59.8,5,62.
     6,3
3150 DATA 64.1,3,69.9,3,71.2,2,7
     2.1,2,76.6,2,103,4
```

Seguito figura 2.

BASIC

DIZ 80. Un disassemblatore interattivo in BASIC per Z80

Un disassemblatore traduce da linguaggio numerico di macchina in linguaggio assemblativo. Esso dovrebbe far parte del corredo del software di base fornito dalle case costruttrici, alla stessa stregua dei compilatori, assemblatori, sistemi operativi, interpreti, ecc. Non è presente, però, su alcuni personal, specialmente su quelli che adottano microprocessori con un formato ed un set d'istruzioni piuttosto complesso come lo Z80.

di **A. Gentile De Blasi**

Il problema più grosso di fronte al quale ci si imbatte nella costruzione di un disassemblatore è quello di distinguere le aree dei dati da quelle delle istruzioni (/1/ e /2/). In particolare in /2/ viene descritto un cross-disassemblatore realizzato in PL/1 su IBM 370. Come lo stesso autore ha fatto notare, l'uso di tale programma è escluso agli utenti di microcomputer e personal computer che hanno una memoria limitata ed usano più comunemente il linguaggio BASIC.

Noi abbiamo voluto, viceversa, soddisfare proprio questa fascia di utenti seguendo un approccio di tipo interattivo. Infatti, il nostro disassemblatore è scritto interamente in BASIC e richiede soltanto una memoria di 16 Kbyte (nella versione ridotta).

Il modo con cui abbiamo risolto il problema di distinguere le aree dei dati da quelle delle istruzioni è stato il seguente. I dati, in genere, sono in una zona distinta del programma, ma se messi nel mezzo di un programma, essi possono essere inseriti solo dopo istruzioni di salto incondizionato.

Allora abbiamo inserito nel nostro programma disassemblatore un controllo su tutti i tipi di istruzioni di salto incondizionato. Cioè, ogni volta che una tale istruzione viene incontrata, dopo averla tradotta regolarmente in simbolico, viene calcolato e quindi scritto sulla stessa linea l'indirizzo (in decimale) a cui tale istruzione punta, dopodichè viene eseguita una istruzione di STOP. A questo punto si può richiedere di continuare a disassemblare dalla locazione di memoria indicata dalla istruzione di salto.

In questo modo si riesce a realizzare, seguendo il flusso del programma attraverso le sue istruzioni di salto, una copertura completa dello stesso. Un problema attualmente non risolto nei disassemblatori è dato dalla presenza di istruzioni di salto con indirizzo implicito (per es. nello Z80 le istruzioni JP (HL), JP (IX+d), JP (IY+d)). Tali istruzioni effettuano il salto ad un indirizzo di memoria che è noto solo all'atto dell'esecuzione del programma.

Noi riteniamo, comunque, che l'approccio interattivo del nostro disassemblatore permetta di aumentare in maniera notevole il grado di copertura delle istruzioni disassemblate, anche in presenza di istruzioni del tipo suddetto.

Esecuzione del programma di disassemblaggio

Dopo aver caricato in memoria il disassemblatore, si dà l'avvio per la sua esecuzione:

```
RUN
? —
```

Questo rimane in attesa dell'indirizzo iniziale e di quello finale (in decimale) dell'area di memoria contenente il programma da disassemblare.

```
RUN
? 49152, 57343
>
```

Nell'esempio è stato scritto come indirizzo iniziale 49152 = C000H e come indirizzo finale 57343 = DFFFH. Questi sono i limiti di memoria EPROM dell'interprete BASIC del Nanocomputer della SGS. In figura 1 è presentato il listing di tale interprete fino alla prima istruzione di salto incondizionato incontrata.

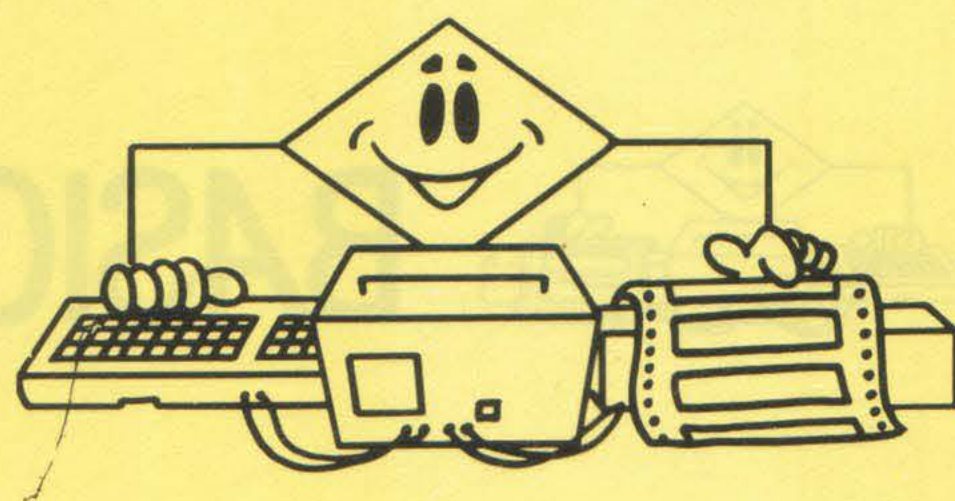
A questa istruzione l'esecuzione si è arrestata fornendo l'indirizzo in decimale della locazione di memoria a cui tale istruzione punta (50145).

A questo punto si può rieseguire il RUN fornendo come indirizzo di partenza 50145:

```
RUN
? 50145, 57343
>
```

Si ottiene così, il listing di figura 2. Dalle figure 1 e 2 si può notare come, anche per le istruzioni di chiamata a subroutine, viene scritto accanto all'istruzione disassemblata l'indirizzo in decimale della subroutine chiamata; questo è molto utile per poter eseguire separatamente il disassemblaggio delle varie subroutine chiamate.

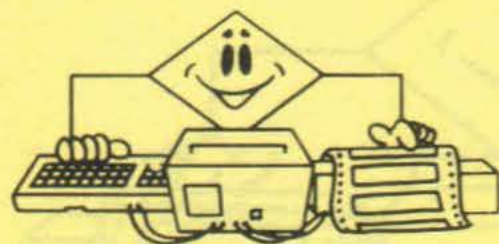
A tal fine è stato introdotto nel programma del disassemblatore anche un controllo sulle istruzioni di ritorno a subroutine, dopo le quali viene eseguito uno STOP.

Come si può notare in figura 1, la prima subroutine chiamata è nella locazione 56812 = DDECH; il

suo disassemblaggio è in figura 3 dove al presentarsi dell'istruzione RET il listing è interrotto automaticamente.
L'indirizzo decimale viene anche stampato accanto alle istruzioni di salto condizionato per facilitare l'interpretazione del programma disassemblato. Inoltre, per le istruzioni di salto di tipo relativo, il programma calcola l'indirizzo assoluto e lo stampa sia in decimale che in esadecimale.

Dump della memoria

Infine, questo disassemblatore permette di fare un DUMP della memoria dalla locazione desiderata, utile per effettuare la stampa delle zone di memoria dove vi sono i dati. Questa viene effettuata riportando l'indirizzo in esadecimale ed il contenuto di 16 byte consecutivi per ogni linea di stampa, come si può vedere in figura 4.

```
D I S A S S E M B L A G G I O  :  49152 (=C000H)

LOCAZIONE   ISTRUZIONE    SIMBOLICO      DEST.
            MACCHINA                     (DEC.)

C000        FB            EI
C001        21 B3 C2      LD HL,C2B3H
C004        01 70 00      LD BC,0070H
C007        11 02 10      LD DE,1002H
C00A        ED B0         LDIR
C00C        EB            EX DE,HL
C00D        F9            LD SP,HL
C00E        CD EC DD      CALL DDECH     56812
C011        CD 0E DE      CALL DE0EH     56846
C014        CD F1 C4      CALL C4F1H     50417
C017        CD 3F CA      CALL CA3FH
C01A        32 05 11
C01D        32 34 1                      50380
                          LD HL,(1067H)
                          LD DE,1234H
                          SBC HL,DE
C098        CD BE D7      CALL D7BEH     55230
C09B        21 AA C0      LD HL,C0AAH
C09E        CD 7D D0      CALL D07DH     53373
C0A1        CD F1 C4      CALL C4F1H     50417
C0A4        CD 11 10      CALL 1011H     4113
C0A7        C3 E1 C3      JP C3E1H       50145
C0AA
```

Figura 1 - Output del disassemblatore. In prima colonna sono stampati gli indirizzi esadecimali delle successive istruzioni disassemblate. In seconda e terza colonna sono riportate le istruzioni in numerico e le corrispondenti istruzioni disassemblate. Nell'ultima colonna è stampato (in decimale) l'indirizzo di destinazione per le istruzioni di trasferimento del controllo (salti e salti a subroutine). Il disassemblaggio viene interrotto al primo salto incondizionato: nell'esempio, alla locazione C047 dove c'è l'istruzione JP C3E1. il disassemblaggio può essere fatto proseguire dalla C3E1H = 50145 (figura 2).

BASIC

```
RUN 6000
? —
```

L'esecuzione del programma di DUMP inizia dalla linea 6000 del programma disassemblatore, per cui, dopo aver effettuato il RUN 6000, il sistema è in attesa degli indirizzi, iniziale e finale (in decimale) della zona di memoria di cui si vuole il DUMP.

```
RUN 6000
? 49152, 49232
>
```

A scopo dimostrativo è stato fatto il DUMP delle stesse locazioni di memoria di figura 1, anche se queste contengono istruzioni e non dati.

Conclusione

Il programma completo del disassemblatore, con DUMP di memoria e calcolo degli indirizzi a cui puntano le istruzioni di salto e di chiamata a subroutine, è scritto interamente in BASIC.
Per venire incontro alle esigenze di utenti di microcomputer aventi una quantità di memoria RAM non superiore a 16 Kbyte è stato realizzato un disassemblatore "ridotto" in cui è stata eliminata la codifica del codice esadecimale in simbolico di alcune istruzioni poco usate (queste sono le istruzioni di shift, di rotazione e di manipolazione di bit).

Con questa versione ridotta, quando una delle istruzioni su citata è incontrata durante la fase di disassemblaggio, viene stampato (come per le altre istruzioni) l'indirizzo esadecimale ed il codi-

Figura 2 - Output del disassemblatore a cominciare dalla locazione di destinazione del primo salto incondizionato (incontrato in figura 1). Si noti come, anche per i salti relativi, viene fornito l'indirizzo di destinazione. Questo viene calcolato a partire dall'indirizzo relativo e stampato sia in esadecimale, per aiutare nell'analisi del programma, che in decimale per eventuali richieste di disassemblaggio a partire da tale locazione.

D I S A S S E M B L A G G I O : 50145 (=C3E1H)

LOCAZIONE	ISTRUZIONE MACCHINA	SIMBOLICO	DEST. (DEC.)
C3E1	CD DD C6	CALL C6DDH	50909
C3E4	AF	XOR A	
C3E5	32 6B 10	LD (106BH),A	
C3E8	32 61 10	LD (1061H),A	
C3EB	32 66 10	LD (1066H),A	
C3EE	3A 51 11	LD A,(1151H)	
C3F1	B7	OR A	
C3F2	C4 DB DE	CALL NZ,DEDBH	57051
C3F5	CD 33 CA	CALL CA33H	51763
C3F8	21 2C C3	LD HL,C32CH	
C3FB	CD 7D D0	CALL D07DH	53373
C3FE	21 FF FF	LD HL,FFFFH	
C401	22 80 10	LD (1080H),HL	
C404	CD C9 C5	CALL C5C9H	50633
C407	38 F5	JR C,F5H C3FEH	50174
C409	CD B5 C7	CALL C7B5H	51125
C40C	3C	INC A	
C40D	3D	DEC A	

ce esadecimale dell'istruzione, dopodichè il programma disassemblatore passa ad esaminare la prossima istruzione. Cioè l'unica differenza con le altre istruzioni è che di queste non viene effettuata la traduzione in simbolico. (v. figura 5). Questo non è un inconveniente molto grave poiché, come si è già detto, tali istruzioni sono molto rare in un generico programma per cui non sarà poi una gran fatica tradurre manualmente il simbolico corrispondente per queste sole istruzioni.

Come si può vedere da figura 5, le istruzioni di indirizzo esadecimale DDDA, DDDC e DDE1 non sono state disassemblate.

Comunque esiste un disassemblatore completo per utenti con memoria maggiore di 16 Kbyte.

Bibliografia

Khalaf Al-Jumaily, S Das Gupta and Dakhil Jerew, "Identification of a program segment from memory dump", Microprocessors and microsystems, Vol. 4 no. 1 Gennaio/Febbraio 80, pp. 11-14.

S. Margarita, "Un cross-disassemblatore (quasi) intelligente", **Bit**, Anno 3 - N° 8/9 - Giugno/Agosto 80, pp. 70-77.

```
D I S A S S E M B L A G G I O     56812 (=DDECH)

LOCAZIONE   ISTRUZIONE          SIMBOLICO           DEST.
            MACCHINA                                (DEC.)

DDEC        3E 0F               LD A,0FH
DDEE        D3 07               OUT (07H),A
DDF0        3E CF               LD A,CFH
DDF2        D3 05               OUT (05H),A
DDF4        D3 06               OUT (06H),A
DDF6        3E 8F               LD A,8FH
DDF8        D3 06               OUT (06H),A
DDFA        3E 50               LD A,50H
DDFC        D3 04               OUT (04H),A
DDFE        3E 03               LD A,03H
DE00        D3 06               OUT (06H),A
DE02        21 0C 10            LD HL,100CH
DE05        7D                  LD A,L
DE06        D3 06               OUT (06H),A
DE08        7C                  LD A,H
DE09        ED 47               LD I,A
DE0B        ED 5E               IM 2
DE0D        C9                  RET
DE0E
```

Figura 3 - Listing della subroutine con indirizzo iniziale 56812. Tale indirizzo è stato ottenuto esaminando un listing precedente (quello di figura 1). Per permettere la stampa delle subroutine, un'ulteriore condizione di arresto del disassemblaggio è costituita dall'istruzione di ritorno al programma chiamante (istruzione RET).

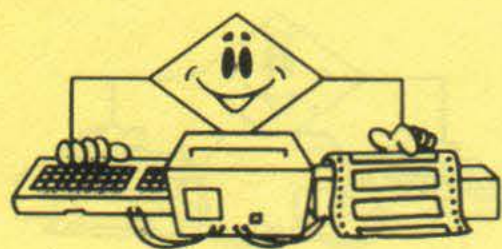
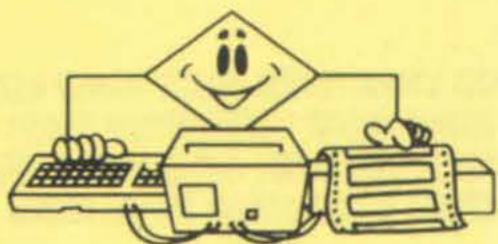
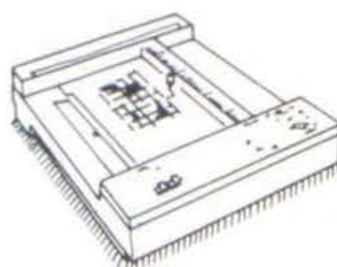



Figura 4 - Dump della memoria. Ogni riga contiene 10H = 16 byte di memoria.

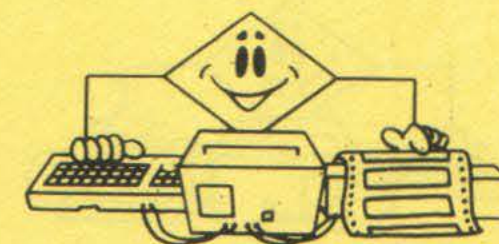
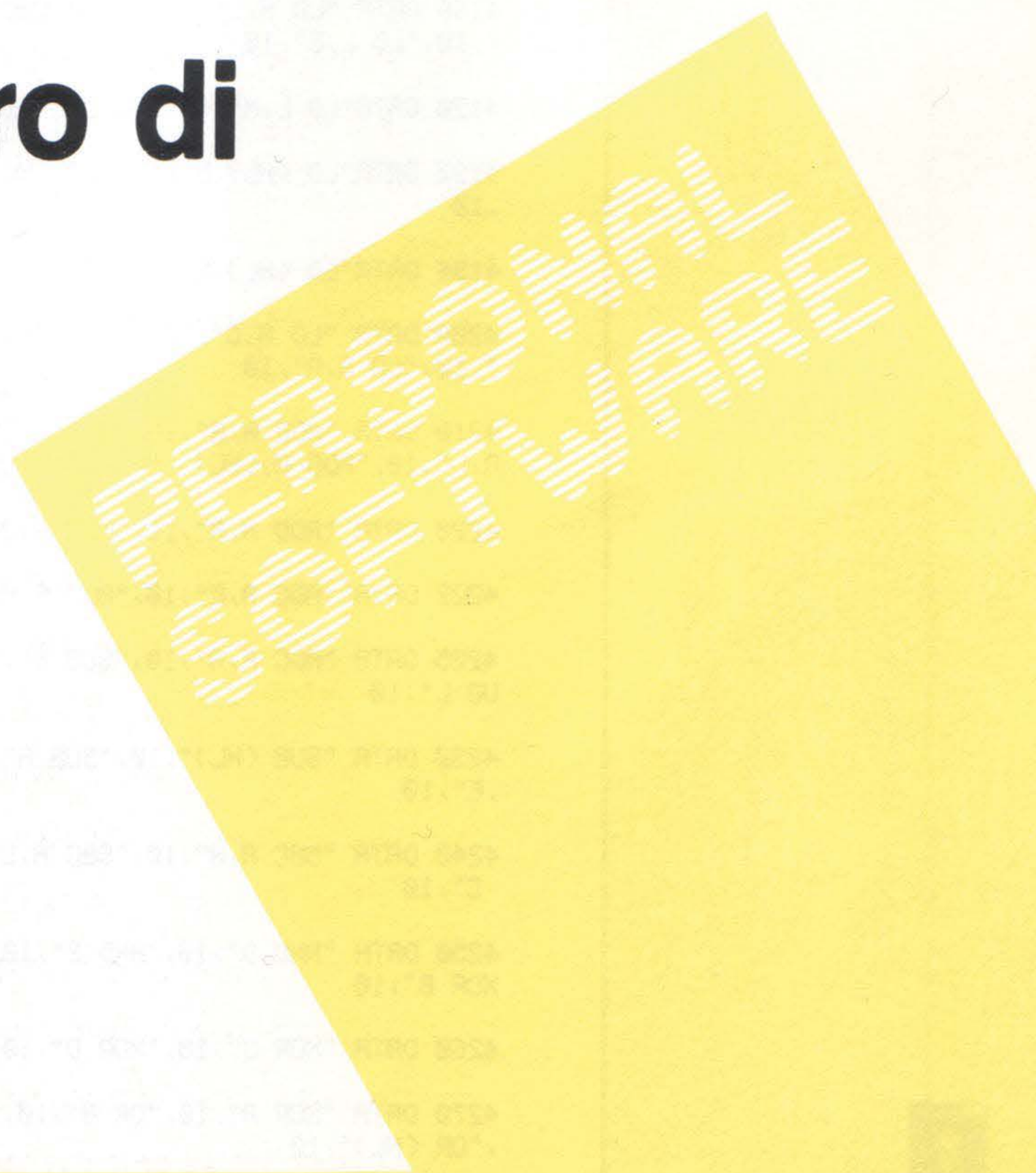
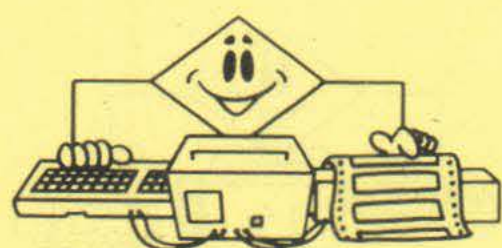
```
C000        FB 21 B3 C2 01 70 00 11 02 10 ED B0 EB F9 CD EC
C010        DD CD 0E DE CD F1 C4 CD 3F CA 32 05 11 32 34 12
C020        21 B8 C0 CD 7D D0 CD F1 C4 21 E0 C0 CD 7D D0 CD
C030        15 C5 38 F5 CD B5 C7 B7 28 1A CD 69 C8 B2 20 E9
C040        7B FE 0F 38 E4 32 64 10 D6 0E 30 FC C6 1C 2F 3C
C050        83
```

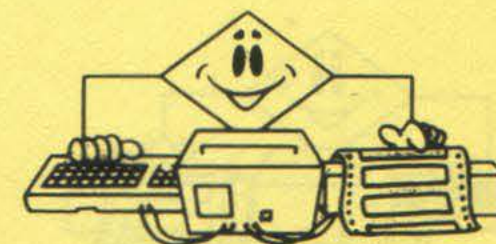
Figura 5 - Esempio di output del disassemblatore ridotto. Si noti come alcune istruzioni, CB 1C, CB 1D e CB CC, benché correttamente riportate nella parte numerica, non sono state disassemblate per mancanza di memoria. Queste istruzioni sono relativamente poco frequenti e possono essere tradotte a mano. Il disassemblatore completo, viceversa, fornisce il listing disassemblato di tutte le istruzioni incontrate.

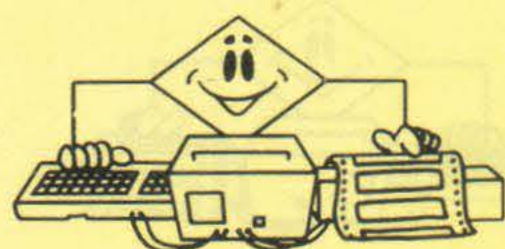
```
D I S A S S E M B L A G G I O  :  56785 (=DDD1H)

LOCAZIONE   ISTRUZIONE          SIMBOLICO                  DEST.
            MACCHINA                                       (DEC.)

DDD1        30 FB               JR NC,FBH       DDCEH      56782
DDD3        2A 7D 10            LDHL,(107DH)
DDD6        7A                  LD A,D
DDD7        D6 12               SUB 12H
DDD9        37                  SCF
DDDA        CB 1C
DDDC        CB 1D
DDDE        3C                  INC A
DDDF        20 F8               JR NZ,F8H       DDD9H      56793
DDE1        CB CC
DDE3        3F                  CCF
DDE4        8C                  ADC A,H
DDE5        20 03               JR NZ,03H       DDEAH      56810
DDE7        22 7B 10            LD  ( 107BH),HL
DDEA        18 C7               JRC7H           DDB3H      56755
DDEC
```

Listato 1 - Il programma DIZ'80.

```
1 GOTO 20000
100 REM
110 N$=HEX$(N)
120 IF LEN(N$)=1 THEN N$="0"+N$
130 RETURN
150 REM
160 N=PEEK (AN):GOSUB 100
170 RETURN
200 REM
210 X=INT(D/256)
220 H$=HEX$(X): X$=HEX$(D-X*256)
230 IF LEN(H$)=1 THEN H$="0"+H$
240 IF LEN(X$)=1 THEN X$="0"+X$
250 H$=H$+X$
260 RETURN
300 IF OP=PP GOTO 340
310 IF OP<PP THEN PP=-1:RESTORE
320 FOR I =PP+1 TO OP
330 READ S$,NB:NEXT I
340 RETURN
400 AN=AN+1:A=A+1:GOSUB 150:LPRINT N$;
410 N3$=N$
420 AN=AN+1:A=A+1:GOSUB 150 :LPRINT N$;
430 RETURN
500 AN=AN+1:A+1:GOSUB 150:LPRINT N$;
510 RETURN
600 FOR I=0 TO OP
610 READ S$:NEXT I
620 RETURN
650 REM
660 IF A<=32767 THEN AN=A:GOTO 680
670 AN=A-65536
680 RETURN
1000 OP=PEEK(AN)
```

Seguito programma DIZ 80.

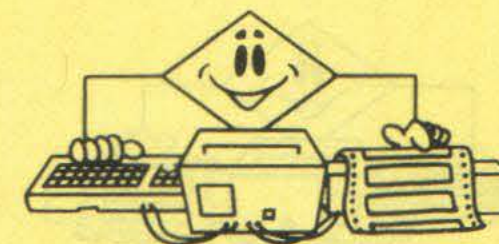
```
1010 IF OP=9 THEN LPRINT,,"ADD ";Z$;",BC":GOTO 20230
1020 IF OP=25 THEN LPRINT,,"ADD ";Z$;",DE":GOTO 20230
1030 IF OP=33 THEN GOSUB 400:LPRINT,,"LD ";Z$;",";N$;N3$;"H":GOTO 20230
1040 IF OP=34 THEN GOSUB 400:LPRINT,,"LD (";N$;N3$;"H)";Z$:GOTO 20230
1050 IF OP=35 THEN LPRINT,,"INC ";Z$:GOTO 20230
1060 IF OP=41 THEN LPRINT,,"ADD ";Z$;",";Z$:GOTO 20230
1070 IF OP=42 THEN GOSUB 400:LPRINT,,"LD ";Z$;",(";N$;N3$;"H)":GOTO 20230
1080 IF OP=43 THEN LPRINT,,"DEC ";Z$:GOTO 20230
1090 IF OP=52 THEN GOSUB 500:LPRINT,,"INC (";Z$;"+";N$;"H)":GOTO 20230
1100 IF OP=53 THEN GOSUB 500:LPRINT,,"DEC (";Z$;"+";N$;"H)":GOTO 20230
1110 IF OP=54 THEN GOSUB 400:LPRINT,,"LD (";Z$;"+";N3$;"H),";N$;"H":GOTO 20230
1120 IF OP=57 THEN LPRINT,,"ADD ";Z$;",SP":GOTO 20230
1130 IF OP=70 THEN GOSUB 500:LPRINT,,"LD B,(";Z$;"+";N$;"H)":GOTO 20230
1140 IF OP=78 THEN GOSUB 500:LPRINT,,"LD C,(";Z$;"+";N$;"H)":GOTO 20230
1150 IF OP=86 THEN GOSUB 500:LPRINT,,"LD D,(";Z$;"+";N$;"H)":GOTO 20230
1160 IF OP=94 THEN GOSUB 500:LPRINT,,"LD E,(";Z$;"+";N$;"H)":GOTO 20230
1170 IF OP=102 THEN GOSUB 500:LPRINT,,"LD H,(";Z$;"+";N$;"H)":GOTO 20230
1180 IF OP=110 THEN GOSUB 500:LPRINT,,"LD L,(";Z$;"+";N$;"H)":GOTO 20230
1190 IF OP=112 THEN GOSUB 500:LPRINT,,"LD (";Z$;"+";N$;"H),B":GOTO 20230
1200 IF OP=113 THEN GOSUB 500:LPRINT,,"LD (";Z$;"+";N$;"H),C":GOTO 20230
1210 IF OP=114 THEN GOSUB 500:LPRINT,,"LD (";Z$;"+";N$;"H),D":GOTO 20230
1220 IF OP=115 THEN GOSUB 500:LPRINT,,"LD (";Z$;"+";N$;"H),E":GOTO 20230
1230 IF OP=116 THEN GOSUB 500:LPRINT,,"LD (";Z$;"+";N$;"H),H":GOTO 20230
1240 IF OP=117 THEN GOSUB 500:LPRINT,,"LD (";Z$;"+";N$;"H),L":GOTO 20230
1250 IF OP=119 THEN GOSUB 500:LPRINT,,"LD (";Z$;"+";N$;"H),A":GOTO 20230
1260 IF OP=126 THEN GOSUB 500:LPRINT,,"LD A,(";Z$;"+";N$;"H)":GOTO 20230
1270 IF OP=134 THEN GOSUB 500:LPRINT,,"ADD A,(";Z$;"+";N$;"H)":GOTO 20230
1280 IF OP=142 THEN GOSUB 500:LPRINT,,"ADC A,(";Z$;"+";N$;"H)":GOTO 20230
1290 IF OP=150 THEN GOSUB 500:LPRINT,,"SUB (";Z$;"+";N$;"H)":GOTO 20230
1300 IF OP=158 THEN GOSUB 500:LPRINT,,"SBC A,(";Z$;"+";N$;"H)":GOTO 20230
1310 IF OP=166 THEN GOSUB 500:LPRINT,,"AND (";Z$;"+";N$;"H)":GOTO 20230
1320 IF OP=174 THEN GOSUB 500:LPRINT,,"XOR (";Z$;"+";N$;"H)":GOTO 20230
1330 IF OP=182 THEN GOSUB 500:LPRINT,,"OR (";Z$;"+";N$;"H)":GOTO 20230
1340 IF OP=190 THEN GOSUB 500:LPRINT,,"CP (";Z$;"+";N$;"H)":GOTO 20230
1350 IF OP=225 THEN LPRINT,,"POP ";Z$:GOTO 20230
1360 IF OP=227 THEN LPRINT,,"EX (SP),";Z$:GOTO 20230
1370 IF OP=229 THEN LPRINT,,"PUSH ";Z$:GOTO 20230
1380 IF OP=233 THEN LPRINT,,"JP (";Z$;")":GOTO 20230
```

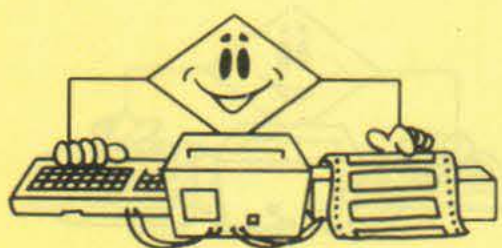
Seguito programma DIZ 80.

```
1390 IF OP=249 THEN LPRINT,,"LD SP,";Z$:GOTO 20230

1510 GOSUB 400:LPRINT:GOTO 20230

1600 IF OP=56 OR OP=24 OR OP=48 OR OP=32 OR OP=40 GOTO 1620

1610 LPRINT:GOTO 20230

1620 R=PEEK(AN)+2

1630 IF R>129 THEN R=R-256

1635 D=R+A-1

1640 GOSUB 200:LPRINT,H$;"H",D:GOTO 20230

1650 IFOP=195 OR OP=218 OROP=250 OR OP=210 OR OP=194 OR OP=242 OR OP=234 OR OP=2
26 OR OP=202 GOTO 1680

1660 IF OP=205 OR OP=220 OR OP=252 OR OP=212 OR OP=196 OR OP=244 OR OP=236 OR OP
=228 OR OP=204 GOTO 1680

1665 GOSUB 650

1670 LPRINT :GOTO 20230

1680 AA=PEEK(AN)*256+PEEK(AN-1):LPRINT ,,AA:GOTO 20230

4050 DATA "NOP",10,"LD BC,",31,"LD (BC),A",10,"INC BC",10,"INC B",10
```

Seguito programma DIZ 80.

```
4060 DATA "DEC B",10,"LD B,",21,"RLCA",10,"EX AF,AF'",10,"ADD HL,BC",10,"LD A,(B
C)",10,"DEC BC",10

4070 DATA "INC C",10,"DEC C",10,"LD C,",21,"RRCA",10,"DJNZ ",21,"LD DE,",31

4080 DATA "LD (DE),A",10,"INC DE",10,"INC D",10,"DEC D",10,"LD D,",21,"RLA",10,"
JR",21,"ADD HL,DE",10,"LD A,(DE)",10,"DEC DE",10

4090 DATA "INC E",10,"DEC E",10,"LD E,",21,"RRA",10,"JR NZ,",21,"LD HL,",31

4100 DATA "LD  (  ),HL",32,"INC HL",10,"INC H",10,"DEC H",10,"LD H,",21,"DAA",10

4104 DATA "JR Z,",21,"ADD HL,HL",10,"LDHL,()   ",32

4110 DATA "DEC HL",10,"INC L",10,"DEC L",10,"LD L,",21,"CPL",10,"JR NC,",21,"LD
SP,",31,"LD  (   ),A",32,"INC SP",10

4120 DATA "INC (HL)",10,"DEC (HL)",10,"LD (HL),",21,"SCF",10,"JR C,",21,"ADD HL,
SP",10

4122 DATA "LD A,()   ",32,"DEC SP",10,"INC A",10

4130 DATA "DEC A",10,"LD A,",21,"CCF",10,"LD B,B",10,"LD B,C",10,"LD B,D",10,"LD
 B,E",10,"LD B,H",10

4140 DATA "LD B,L",10,"LD B,(HL)",10,"LD B,A",10,"LD C,B",10,"LD C,C",10,"LD C,D
",10,"LD C,E",10,"LD C,H",10,"LD C,L",10

4150 DATA "LD C,(HL)",10,"LD C,A",10,"LD D,B",10,"LD D,C",10,"LD D,D",10,"LD D,E
",10,"LD D,H",10,"LD D,L",10

4160 DATA "LD D,(HL)",10,"LD D,A",10,"LD E,B",10,"LD E,C",10,"LD E,D",10,"LD E,E
",10,"LD E,H",10,"LD E,L",10

4170 DATA "LD E,(HL)",10,"LD E,A",10,"LD H,B",10,"LD H,C",10,"LD H,D",10,"LD H,E
",10,"LD H,H",10

4180 DATA "LD H,L",10,"LD H,(HL)",10,"LD H,A",10,"LD L,B",10,"LD L,C",10,"LD L,D
",10,"LD L,E",10

4190 DATA"LD L,H",10,"LD L,L",10,"LD L,(HL)",10,"LD L,A",10

4192 DATA"LD (HL),B",10,"LD (HL),C",10,"LD (HL),D",10,"LD (HL),E",10,"LD (HL),H"
,10

4194 DATA"LD (HL),L",10,"HALT",10,"LD (HL),A",10,"LD A,B",10

4200 DATA "LD A,C",10,"LD A,D",10,"LD A,E",10,"LD A,H",10,"LD A,L",10,"LD A,(HL)
",10,"LD A,A",10

4210 DATA "ADD A,B",10,"ADD A,C",10,"ADD A,D",10,"ADD A,E",10,"ADD A,H",10,"ADD
A,L",10,"ADD A,(HL)",10

4220 DATA "ADD A,A",10,"ADC A,B",10,"ADC A,C",10,"ADC A,D",10

4222 DATA "ADC A,E",10,"ADC A,H",10,"ADC A,L",10,"ADC A,(HL)",10

4225 DATA "ADC A,A",10,"SUB B",10,"SUB C",10,"SUB D",10,"SUB E",10,"SUB H",10,"S
UB L",10

4230 DATA "SUB (HL)",10,"SUB A",10,"SBC A,B",10,"SBC A,C",10,"SBC A,D",10,"SBC A
,E",10

4240 DATA "SBC A,H",10,"SBC A,L",10,"SBC A,(HL)",10,"SBC A,A",10,"AND B",10,"AND
 C",10

4250 DATA "AND D",10,"AND E",10,"AND H",10,"AND L",10,"AND (HL)",10,"AND A",10,"
XOR B",10

4260 DATA "XOR C",10,"XOR D",10,"XOR E",10,"XOR H",10,"XOR L",10,"XOR (HL)",10

4270 DATA "XOR A",10,"OR B",10,"OR C",10,"OR D",10,"OR E",10,"OR H",10,"OR L",10
,"OR (HL)",10
```

```
4280 DATA "OR A",10,"CP B",10,"CP C",10,"CP D",10,"CP E",10,"CP H",10,"CP L",10,
"CP (HL)",10,"CP A",10

4290 DATA "RET NZ",10,"POP BC",10,"JP NZ,",31,"JP ",31,"CALL NZ,",31,"PUSH BC",1
0,"ADD A,",21

4300 DATA "RST 0",10,"RET Z",10,"RET",10,"JP Z,",31,"CB",0,"CALL Z,",31,"CALL ",
31,"ADC A,",21,"RST 8",10

4310 DATA "RET NC",10,"POP DE",10,"JP NC,",31,"OUT (),A",22,"CALL NC,",31,"PUSH
DE",10,"SUB ",21

4320 DATA "RST 10H",10,"RET C",10,"EXX",10,"JP C,",31,"INA,()  ",22,"CALL C,",31
,"DD",0,"SBC A,",21

4330 DATA "RST 18H",10,"RET PO",10,"POP HL",10,"JP PO,",31,"EX (SP),HL",10,"CALL
 PO,",31

4340 DATA "PUSH HL",10,"AND",21,"RST 20H",10,"RET PE",10,"JP (HL)",10,"JP PE,",3
1,"EX DE,HL",10

4350 DATA "CALL PE,",31,"ED",0,"XOR ",21,"RST 28H",10,"RET P",10,"POP AF",10,"JP
 P,",31,"DI",10,"CALL P,",31

4360 DATA "PUSH AF",10,"OR ",21,"RST 30H",10,"RET M",10,"LD SP,HL",10,"JP M,",31
,"EI",10,"CALL M,",31

4361 DATA "FD",0

4370 DATA"CP ",21,"RST 38H",10
```

Seguito programma DIZ 80.

BASIC

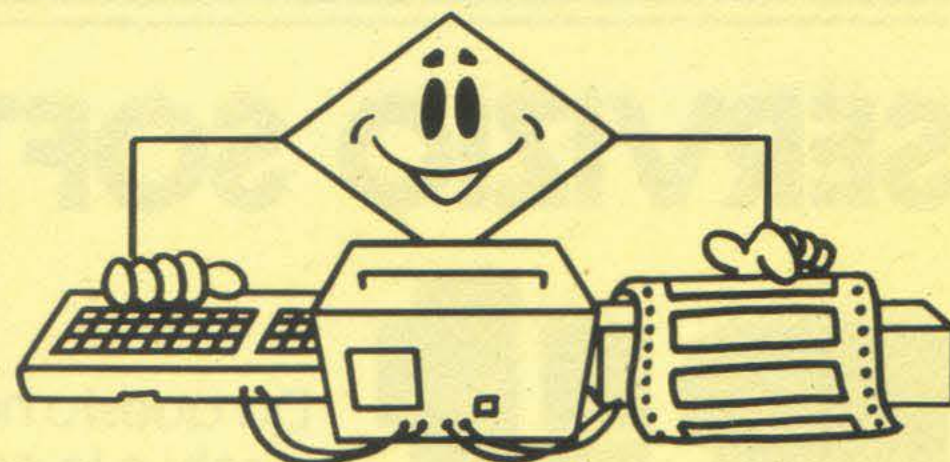
Seguito programma DIZ 80.

```
6000 REM
6010 INPUT A,AM
6020 Z=0:ZM=AM-A+1
6030 LPRINT
6050 D=A:GOSUB 200:LPRINT H$,
6060 FOR Y=1 TO 16
6065 GOSUB 650
6070 GOSUB 150:LPRINT N$;" ";
6080 Z=Z+1
6090 IF Z>=ZM GOTO 6120
6100 A=A+1:NEXT Y
6110 GOTO 6030
6120 LPRINT :END
20000 REM
20010 INPUT A,AM
20012 LPRINT:LPRINT:LPRINT:LPRINT:LPRINT:LPRINT
20015 LPRINT"D I S A S S E M B L A G G I O   :  ";A;
20020 CB=203: DD=221: ED=237: FD=253
20021 TZ=-1:PP=-1
20022 IF A<=32767 THEN AN=A:GOTO 20030
20024 AN=A-65536
20026 D=A: GOSUB 200: LPRINT " (=";H$;"H)"
20027 LPRINT:LPRINT
20028 LPRINT "LOCAZIONE","ISTRUZIONE",,"SIMBOLICO",,"DEST."
20029 LPRINT,"MACCHINA",,,,"(DEC.)":LPRINT
20030 D=A: GOSUB 200: LPRINT H$,
20033 PP=TZ
20035 IF PP=195 OR PP=24 OR  PP=201 THEN STOP
20040 GOSUB 150: LPRINT N$; " ";
20050 OP=PEEK(AN):TZ=OP
20060 IF OP=CB GOTO 20260
20070 IF OP=ED OR OP=DD OR OP=FD GOTO 20290
20102 GOSUB 300
20110 L=INT(NB/10): T=NB-L*10
20130 ON L GOTO 20140,20150,20190
20140 LPRINT ,,S$:GOTO 20230
20150 AN=AN+1 :A=A+1 :GOSUB 150:LPRINT N$;
```

Seguito programma DIZ 80.

```
20160 IF T=1 THEN LPRINT ,,S$;:GOSUB 150:LPRINT N$;"H";:GOTO1600
20170 LPRINT,,LEFT$(S$,5);:GOSUB 150:LPRINT N$;"H";
20180 LPRINT RIGHT$(S$,3):GOTO 20230
20190 AN=AN+1: A=A+1:GOSUB 150 :LPRINT N$;" ";:Y$=N$
20200 AN=AN+1:A=A+1: GOSUB 150:LPRINT N$;
20210 IF T=1 THEN LPRINT,,S$;N$;Y$;"H";:GOTO 1650
20220 LPRINT,,LEFT$(S$,6);N$;Y$;"H";RIGHT$(S$,4):GOTO 20230
20230 AN=AN+1:A=A+1
20240 IF A>AM THEN END
20250 GOTO 20030
20260 AN=AN+1:A=A+1
20270 GOSUB 150:LPRINT N$;" ";
20280 LPRINT:GOTO20230
20290 AN=AN+1:A=A+1:GOSUB150:LPRINT N$;" ",
20292 N2$=N$
20294 IF OP=ED GOTO 20700
20296 IF OP=DD THEN Z$="IX":GOTO 1000
20298 Z$="IY":GOTO 1000
20700 OP=PEEK(AN)
20710 IF OP=64 THEN LPRINT,,"IN B,(C)":GOTO20230
20720 IF OP=65 THEN LPRINT,,"OUT (C),B":GOTO 20230
20730 IF OP=66 THEN LPRINT,,"SBC HL,BC":GOTO 20230
20740 IF OP=67 THEN GOSUB 400:LPRINT,,"LD (";N$;N3$;"H),BC":GOTO 20230
20750 IF OP=68 THEN LPRINT,,"NEG":GOTO 20230
20760 IF OP=69 THEN LPRINT,,"RETN":GOTO 20230
20770 IF OP=70 THEN LPRINT,,"IM 0":GOTO 20230
20780 IF OP=71 THEN LPRINT,,"LD I,A":GOTO 20230
20790 IF OP=72 THEN LPRINT,,"IN C,(C)":GOTO 20230
20800 IF OP=73 THEN LPRINT,,"OUT (C),C":GOTO 20230
20810 IF OP=74 THEN LPRINT,,"ADC HL,BC":GOTO 20230
20820 IF OP=75 THEN GOSUB 400:LPRINT,,"LD BC,(";N$;N3$;"H)":GOTO 20230
20830 IF OP=77 THEN LPRINT,,"RETI":GOTO 20230
20840 IF OP=79 THEN LPRINT,,"LD R,A":GOTO 20230
20850 IF OP=80 THEN LPRINT,,"IN D,(C)":GOTO 20230
20860 IF OP=81 THEN LPRINT,,"OUT (C),D":GOTO 20230
20870 IF OP=82 THEN LPRINT,,"SBC HL,DE":GOTO 20230
20880 IF OP=83 THEN GOSUB 400:LPRINT,,"LD (";N$;N3$;"H),DE":GOTO 20230
20890 IF OP=86 THEN LPRINT,,"IM 1":GOTO 20230
```

BASIC

Seguito programma DIZ 80.

```
20900 IF OP=87 THEN LPRINT,,"LD A,I":GOTO 20230
20910 IF OP=88 THEN LPRINT,,"IN E,(C)":GOTO 20230
20920 IF OP=89 THEN LPRINT,,"OUT (C),E":GOTO 20230
20930 IF OP=90 THEN LPRINT,,"ADC HL,DE":GOTO 20230
20940 IF OP=91 THEN GOSUB 400:LPRINT,,"LD DE,(";N$;N3$;"H)":GOTO 20230
20950 IF OP=94 THEN LPRINT,,"IM 2":GOTO 20230
20960 IF OP=95 THEN LPRINT,,"LD A,R":GOTO 20230
20970 IF OP=96 THEN LPRINT,,"IN H,(C)":GOTO 20230
20980 IF OP=97 THEN LPRINT,,"OUT (C),H":GOTO 20230
20990 IF OP=98 THEN LPRINT,,"SBC HL,HL":GOTO 20230
21000 IF OP=103 THEN LPRINT,,"RRD":GOTO 20230
21010 IF OP=104 THEN LPRINT,,"IN L,(C)":GOTO 20230
21020 IF OP=105 THEN LPRINT,,"OUT (C),L":GOTO 20230
21030 IF OP=106 THEN LPRINT,,"ADC HL,HL":GOTO 20230
21040 IF OP=111 THEN LPRINT,,"RLD":GOTO 20230
21050 IF OP=114 THEN LPRINT,,"SBC HL,SP":GOTO 20230
21060 IF OP=115 THEN GOSUB 400:LPRINT,,"LD C";N$;N3$;"H),SP":GOTO 20230
21070 IF OP=120 THEN LPRINT,,"IN A,(C)":GOTO 20230
21080 IF OP=121 THEN LPRINT,,"OUT (C),A":GOTO 20230
21090 IF OP=122 THEN LPRINT,,"ADC HL,SP":GOTO 20230
21100 IF OP=123 THEN GOSUB 400:LPRINT,,"LD SP,(";N$;N3$;"H)":GOTO 20230
21110 IF OP=160 THEN LPRINT,,"LDI":GOTO 20230
21120 IF OP=161 THEN LPRINT,,"CPI":GOTO 20230
21130 IF OP=162 THEN LPRINT,,"INI":GOTO 20230
21140 IF OP=163 THEN LPRINT,,"OUTI":GOTO 20230
21150 IF OP=168 THEN LPRINT,,"LDD":GOTO 20230
21160 IF OP=169 THEN LPRINT,,"CPD":GOTO 20230
21170 IF OP=170 THEN LPRINT,,"IND":GOTO 20230
21180 IF OP=171 THEN LPRINT,,"OUTD":GOTO 20230
21190 IF OP=176 THEN LPRINT,,"LDIR":GOTO 20230
21200 IF OP=177 THEN LPRINT,,"CPIR":GOTO 20230
21210 IF OP=178 THEN LPRINT,,"INIR":GOTO 20230
21220 IF OP=179 THEN LPRINT,,"OTIR":GOTO 20230
21230 IF OP=184 THEN LPRINT,,"LDDR":GOTO 20230
21240 IF OP=185 THEN LPRINT,,"CPDR":GOTO 20230
21250 IF OP=186 THEN LPRINT,,"INDR":GOTO 20230
21260 IF OP=187 THEN LPRINT,,"OTDR":GOTO 20230
```

CASIO

Immersione rapida! Preparare i siluri!

Ecco a voi un gioco veramente originale che potrete programmare sulla vostra Casio FX 702P.

```
LIST #4
  2 WAIT 20:PRT CSR
     6;"≤U-BOAT≥":G
    OTO 5
  3 X=T
  5 PRT "BEST SCORE
    =";X;" PUNTI *"
    ;Z$;"*"
 10 H=0:N=0
 20 IF N≥7 THEN 270
 30 WAIT 10:PRT "SI
    LURI:";7-N
 34 C=INT (RAN#*15+
    1):G$="":D$="":
    K=19:R=19
 36 IF RAN#≥.5;G$="
    *":K=INT (RAN#*
    17+1)
 38 IF H>40;D$="*":
    R=INT (RAN#*17+
    1)
 40 A=RAN#
 50 IF A≤.35;PRT "L
    ANCIA":B=0:$="-
    ":GOTO 99
 60 IF (A-.36)*(A-.
    65)<0:PRT "CACC
    IA":B=.3:$="≥":
    GOTO 99
 70 PRT "PORTAEREI"
    :B=.5:$="+"
 99 FOR I=0 TO 19
103 WAIT B
105 PRT CSR I;$;CSR
     C;"(+)";CSR K;
    G$;CSR R;D$
108 IF (C-K)*(C+2-K
    )≤0 THEN 600
110 IF (C-R)*(C+2-R
    )≤0 THEN 600
115 IF KEY="D";C=C-
    1
120 IF KEY="F";C=C+
    1
125 IF KEY="C";C=C-
    3
130 IF KEY="G";C=C+
    3
135 IF C>16;C=0:GOT
    O 103
140 IF C<0;C=16:GOT
    O 103
145 IF KEY="E" THEN
     500
150 IF N>6 THEN 270
160 NEXT I
260 GOTO 20
270 IF H>X THEN 700
300 WAIT 20:PRT "*G
    AME OVER*";H;"
    PUNTI"
310 STOP :GOTO 5
500 WAIT 5:N=N+1:PR
    T "FUORI";N;" !
    "
505 IF I=C+1;WAIT 1
    5:PRT "BOOOM!",
    "TEMPO:";I:GSB
    800:GOTO 515
510 WAIT 15:PRT "WO
    OOSHHH!":WAIT 8
    :GOTO 150
515 IF $="+";W=20+Y
    :GOTO 530
520 IF $="≥";W=15+Y
    :GOTO 530
525 W=25+Y
530 PRT INT (W-I/2)
    ;" PUNTI / TOT=
    ";H+INT (W-I/2)
535 H=H+INT (W-I/2)
    :T=H:GOTO 20
600 WAIT 20:PRT "MI
    NA!!","BOOOM!!"
610 WAIT 10:PRT ".
    GLU...GLU...GLU
    .......":GOTO 2
    70
700 WAIT 20:PRT "WO
    OW! FANTASTICO!
    ,";H;" PUNTI!"
705 INP "NOME (<6 L
    ETTERE)",Z$:IF
    LEN(Z$)>6 THEN
    705
710 PRT Z$;" HAI IL
     BEST SCORE!!":
    STOP :GOTO 3
800 Y=0
805 IF I=1:PRT "BON
    US 40 PUNTI !":
    Y=40
810 RET
```

Figura 1 - Il listato BASIC.

di **G. Albertini**

Potrete sfruttare al massimo il vostro computer tascabile per giocare ai livelli dei veri video-game.
Dovete infatti giocare astutamente la vostra riserva di sette siluri per affondare le navi nemiche che scorrono sul display e che state puntanto attraverso il mirino del periscopio del vostro sommergibile.

Naturalmente potete puntare le camere di lancio con un semplice tocco sui tasti della vostra Casio e inseguire così il bersaglio. Con un semplice tocco farete anche partire il siluro. Ma attenzione! Le navi affonderanno solo se al momento in cui sparerete un siluro, si troveranno esattamente sopra la croce del mirino. Il punteggio che otterrete sarà relativo al tipo di imbarcazione colpita e sarà tanto più alto quanto minore sarà il tempo impiegato ad affondarla. Avrete inoltre una sorpresa se sarete dei veterani del mare e riuscirete a centrare il bersaglio nel tempo minimo (tempo 1)!

Dovrete fare anche i conti con le mine che galleggiano minacciose a mezz'acqua, innocue per le navi nemiche ma micidiali per voi! Una di esse compare casualmente durante tutta la partita, mentre l'altra rimarrà se supererete i 40 punti. Naturalmente dovrete cercare di ottenere il punteggio più alto possibile e se supererete il BEST SCORE ottenuto fino a quel momento, il vostro nome comparirà all'inizio di ogni partita a memoria della vostra bravura.

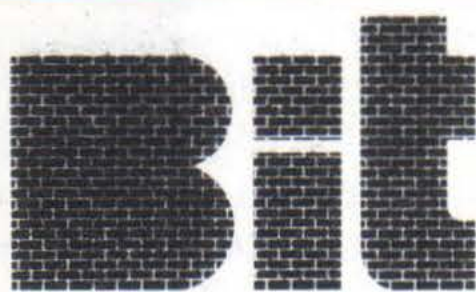
Come manovrare il vostro U-Boat? È molto semplice; avete sulla tastiera 5 tasti a vostra disposizione per le manovre, e cioè quelli corrispondenti alle lettere C, D, E, F, G.
Dovrete giocare a due mani in questo modo: premendo il tasto C vi muoverete verso sinistra in maniera veloce, con il tasto D sempre verso sinistra, ma più lentamente. Analogamente verso destra con i tasti G (veloce) e F (lento). Premendo E farete invece partire il siluro.
Se il vostro mirino arriverà all'estremità sinistra del display ricomparirà a destra e viceversa.

Ora non rimane che programmare la vostra FX (attenzione al n° di passi, ben 1009) e iniziare una emozionante partita! Buon divertimento e attenti alle mine!

SERVIZIO SOFTWARE

Da questo numero Bit propone ai propri lettori i dischi o le cassette dei programmi pubblicati. I programmi, provati e garantiti, sono di immediato utilizzo.

Bit n°	Programma	Sistema	Prezzo	Codice	Supporto
38	Gioco della scimmia Spaccamattoni	VIC 20	15.000	VI381A	Cassetta
38	Text-Editor	PET 3032	20.000	PE381B PE382B	Cassetta Disco
38	Planel	Apple II	20.000	AP382C	Disco
39	Rompicapo di Rubik	CBM 4032 CBM 3032	15.000 20.000	PE391A PE392B	Cassetta Disco
39	Breakout	CBM 3032	20.000	PE393A	Cassetta
40	Reporter	Apple II	20.000	AP402C	Disco
41	Discover	Apple II	20.000	AP412C	Disco
42	Apple-Chef	Apple II	20.000	AP422C	Disco
42	Provariflessi	VIC 20	15.000	VI421A	Cassetta
45	Tiny FORTH	Apple II	35.000	AP452A	Disco
45	Alì Babà	ZX Spectrum	15.000	SP451B	Cassetta
45	1X2	PET 3032	15.000	PE451C	Cassetta

SERVIZIO SOFTWARE

SERVIZIO SOFTWARE

SERVIZIO SOFTWARE

Per richiedere i programmi in contrassegno, pagando direttamente al postino la cifra indicata, inviare il seguente tagliando

Spedire in busta chiusa a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Inviatemi i seguenti nastri e/o dischi con i programmi pubblicati su Bit.

GRUPPO EDITORIALE JACKSON

Cod. ☐☐☐☐☐☐ **a L.** .... **Cognome** ....

Cod. ☐☐☐☐☐☐ **a L.** .... **Nome** ....

Cod. ☐☐☐☐☐☐ **a L.** .... **Indirizzo** ....

Cod. ☐☐☐☐☐☐ **a L.** .... **CAP** .... **Città** ....

Spese postali (contributo fisso) L. 2.000

TOTALE L. ....

che pagherò al postino alla consegna del pacco.

Firma ....